UNIVERSITÀ DEGLI STUDI DI PALERMO
DIPARTIMENTO DELL' INNOVAZIONE INDUSTRIALE E DIGITALE
INGEGNERIA CHIMICA, GESTIONALE, INFORMATICA, MECCANICA
Dottorato in Ingegneria dell'Innovazione Tecnologica – Ingegneria Informatica

SOFTWARE ARCHITECTURES FOR HUMAN-MACHINE INTERACTION USING NATURAL LANGUAGE

SSD: ING-INF/05

TESI DI
ING. GIUSEPPE TIRONE

CO-TUTOR
PROF. ROBERTO PIRRONE

DOTT. ING. ARIANNA PIPITONE

COORDINATORE DEL DOTTORATO
CH.MO PROF. ANTONIO CHELLA

TUTOR
CH.MO PROF. EDOARDO ARDIZZONE

XXX CICLO

DOTTORATO

Ph.D. Thesis

Giuseppe Tirone

February 23, 2018

Software Architectures For Human-Machine Interaction Using Natural Language

Submitted in fulfillment of the requirements for the degree of
Philosophical Doctor of Technological Innovation Engineering
SSD: ING-INF/05
Università degli Studi di Palermo, 2018

Giuseppe Tirone

Dipartimento dell'Innovazione Industriale e Digitale (DIID)
Ingegneria Chimica, Gestionale, Informatica, Meccanica
Università degli Studi di Palermo

«When we understand the brain,
then humanity will have understood itself»

«Quando capiremo il cervello,
allora l'umanità avrà capito se stessa»

(Rafael Yuste)

«We are what we are because our brains are basically
chemical machines, rather than electric»

«Siamo ciò che siamo perchè i nostri cervelli sono
fondamentalmente macchine chimiche, più che elettriche»

(Richard F. Thompson)

Sommario

Il linguaggio naturale rappresenta un sistema di comunicazione a carattere inferenziale in opposizione ai sistemi di comunicazione a codice che non prevedono una forma di ragionamento intelligente da parte del ricevente, ma si basano sul riconoscimento di patterns dell'informazione. In un sistema di comunicazione di tipo inferenziale, infatti, si parte dal presupposto che il ricevente abbia una certa "intelligenza" e sia, quindi, capace di comprendere, elaborare ed inferire il contenuto informativo di una comunicazione attraverso ragionamenti su un background di conoscenze (come modelli di mondo e di linguaggio) condivisi sia dalla sorgente che dal destinatario.

L'attività di ricerca, svolta nell'ambito dei tre anni di corso del Dottorato e presentata in questo lavoro di tesi, pone le sue radici su questi presupposti trovando collocazione in quello specifico ambito dell'Intelligenza Artificiale, quale è la Human-Computer Interaction dialogica. Più dettagliatamente, l'obiettivo principale di tale percorso è stato la realizzazione di una architettura dedicata all'elaborazione del linguaggio naturale umano attraverso tecniche simboliche di inferenza semantica.

Nella prima parte verrà introdotta la tematica di ricerca e le problematiche affrontate. Verranno presentate le principali motivazioni e le criticità connesse al tema affrontato, evidenziando pregi e debolezze degli attuali sistemi che cos-

tituiscono lo stato dell'arte.

Alla luce di tali studi si è scelto di proseguire il percorso approfondendo le tecniche simboliche per l'elaborazione del linguaggio naturale andando a misurare l'efficacia e l'efficienza di tali tecniche in contrapposizione ai sistemi di Information Retrieval (IR) più diffusi, per lo più basati su approcci statistici. Questi ultimi approcci, basandosi per lo più su tecniche di analisi frequentistica, non sono in grado, da un punto di vista formale, di catturare il significato veicolato dal testo, sebbene riescano comunque a garantire un'elevata affidabilità dei risultati, facendo percepire all'utente la sensazione di "essere stato compreso". Alla luce di queste considerazioni verrà presentato QuASIt, il modello proposto, il quale si propone di gestire i due più importanti task nell'ambito del Natural Language Processing (NLP): la comprensione del linguaggio naturale (Understanding) e la produzione dello stesso (Production). Il framework conseguentemente descritto implementa i due task precedentemente indicati, ispirandosi ai processi cognitivi di comprensione ed espressione del parlato. Tale substrato consente la realizzazione di componenti software dedicati a diversi scenari applicativi, quali: annotazione semantica, attività di comprensione di testo non strutturato, analisi di social media, interazione uomo-robot in linguaggio naturale.

Successivamente alla presentazione del modello, verranno illustrate una serie di applicazioni, sviluppate al fine di testare l'efficacia e l'efficenza dello stesso. In particolare, verrà descritto il generico sistema di QA (Question-Answering) basato sul modello cognitivo qui proposto. Successivamente verrà descritta l'applicazione di tale sistema su altri specifici ambiti applicativi. Nel dettaglio è stato valutato sperimentalmente l'uso del sistema come interfaccia dialogica in grado di rispondere a domande su dominio aperto, utilizzando basi di conoscenza strutturate sotto forma di ontologie o testo di supporto non strutturato e rispondendo sia con risposta libera, che a scelta tra un insieme di risposte. Tali test sono stati condotti, sia in lingua italiana che in inglese, in modo da evidenziare l'efficacia del modello cognitivo al variare del dominio linguistico. Un'altra applicazione è consistita nell'uso del sistema come selezionatore di FAQ a partire da domanda libera da parte dell'utente. Infine verrà descritta un'applicazione dell'architettura per l'analisi del testo informale in un tweet al fine dell'identificazione di entità e del linking delle stesse su ontologie.

Il modello e le applicazioni descritte, sono state oggetto di pubblicazione su diverse conferenze internazionali ed i risultati sperimentali ottenuti, che verranno illustrati nella parte finale di questo elaborato, hanno dimostrato l'efficacia e l'efficienza di tale modello.

Abstract

Natural language represents an inferential communication system as opposed to coding communication systems that do not require a form of intelligent reasoning on the part of the recipient, but are based on the recognition of information patterns. Indeed, in an inferential communication system, we start from the assumption that the receiver has a certain “intelligence” and is therefore able to understand, process and infer the information content of a communication through reasoning on a background of knowledge (as models of world and language) shared by both the source and the recipient.

The research activity, carried out within the three years of the PhD program and presented in this dissertation, has its roots on these assumptions finding a place in that specific area of Artificial Intelligence, which is the dialogical Computer-Human Interaction. More specifically, the main objective of this research work has been the realization of an architecture dedicated to the elaboration of human natural language through symbolic techniques of semantic inference.

In the first part, the research topic and the problems addressed will be introduced. The main motivations and criticalities related to the topic will be presented, highlighting the strengths and weaknesses of the current systems that constitute the state of the art.

Grounding on these studies, it was decided to continue the research activity by deepening the symbolic techniques for the processing of natural language, measuring the effectiveness and efficiency of these techniques as opposed to the most widespread information retrieval systems (IR), mostly based on statistical approaches. These approaches, mostly based on frequentistic analysis techniques, are not able, from a formal point of view, to capture the meaning conveyed by the text, although they still manage to guarantee a high reliability of the results, making the user perceive the feeling of "being understood".

In light of these considerations, QuASIt will be presented, the proposed architectural model, which proposes to manage the two most important tasks in the field of Natural Language Processing (NLP): the *understanding* of natural language and its *production*. The framework described below implements the two previously mentioned tasks, inspired by the cognitive processes of comprehension and expression of speech. This substrate allows the creation of software components dedicated to different application scenarios, such as: semantic annotation, non-structured text comprehension activity, social media analysis, human-robot interaction in natural language.

Following the presentation of the model, a series of applications will be illustrated. These are developed in order to test the effectiveness and efficiency of the proposed architecture. In particular, the generic QA (Question-Answering) system based on the cognitive model proposed here will be described. The application of this system will then be described on other specific application areas. In detail, the use of the system as a dialogical interface was able to respond to open domain questions, using knowledge bases structured in the form of ontologies or unstructured support text, and responding either with free response or with a choice between a set of answers. These tests were conducted, both in Italian and in English, in such a way to highlight the effectiveness of the cognitive model as the linguistic domain varies. Another application was the use of the system as a FAQ selector starting from a free application by the user. Finally, an architecture application will be described for the analysis of informal text in a tweet in order to identify entities and link them on ontologies.

The model and the applications described, have been published in various international conferences and the experimental results obtained, which will be illustrated in the final part of this paper, have demonstrated the effectiveness and efficiency of this model.

Ringraziamenti

Il lavoro che vi apprestate a leggere rappresenta, per voi che lo leggerete, il sunto di un lavoro di Ricerca svolto nell'arco di un triennio da uno studente ricercatore. In realtà, per chi lo ha scritto, rappresenta la concretizzazione di quella che inizialmente considerava un'idea, il frutto di una "ingegnosa fantasia", che lo ha portato a leggere, documentarsi e confrontarsi con la letteratura scientifica, con coloro che sono arrivati prima di lui e con coloro che, in laboratorio e nella vita, sono stati a fianco a lui. In sintesi rappresenta il vero significato della Ricerca: far diventare realtà ciò che realtà, prima di allora, non era.

Come si sa, nel mondo, non esistono vette raggiungibili attraverso strade in discesa, anzi occorrerÃ sempre percorrere sentieri irti e pieni di insidie. Ciò che mi ha dato la forza di raggiungere questo traguardo è stata la consapevolezza di non essere mai stato solo perché al mio fianco ho avuto una donna di quelle che qualsiasi uomo aspira ad avere accanto e che io ho la grazia di avere, tanto dolce quanto forte, la mia fidanzata, Adriana. Lei non ha smesso nemmeno per un attimo, insieme a suo fratello Attilio ed ai suoi genitori, di incoraggiarmi. Insieme a lei, ringrazio i miei genitori, i miei scudi e la mia corazza, che mi hanno protetto non facendomi mancare mai il loro sostegno sotto tutti i punti di vista. Una grande famiglia è una grande benedizione e per questo non posso che ringraziare mia sorella Cinzia e Domenico che hanno contribuito in questi anni a renderla più numerosa e allegra regalandomi le mie due splendide nipotine: Maria Cristina e Sofia, coloro che, hanno reso meno "teoriche" e più "sperimentali", le mie giornate durante i miei periodi di studio a casa.

I momenti di crescita culturale in questi tre anni sono stati molteplici, ma non posso che sottolinearne uno in particolare, i mesi trascorsi alla Vrije Universiteit di Brussels in cui ho conosciuto tutor e colleghi di cui porterò per sempre con me uno splendido ricordo. A cominciare dal grandissimo Professore Luc Steels che, insieme alla Professoressa Katrien Beuls, a Remi Van Trijp e a Paul Van Eecke, mi hanno accompagnato in questa esperienza di crescita. Insieme a loro, vorrei ringraziare tutti i colleghi dell'AI Lab che mi sono stati a fianco in questa esperienza, facendomi sentire a Brussels come a casa. Grazie: Piera Filippi, Yannick Jadoul, Michael Anslow, Heikki Rasilo, Bill Thompson, Andrea Ravignani, Joan Loeckx e spero di non dimenticare altri Labsters (in caso è colpa della mancanza di birra/vino caldo!).

Infine, ma ovviamente non per importanza, vorrei ringraziare coloro con cui in questi tre anni ho vissuto più tempo che con qualsiasi altra persona, a cominciare dai miei tutor Proff. Edoardo Ardizzone e Roberto Pirrone, uomini di Scienza che, sempre disponibili al dialogo, mi hanno accompagnato in questo traguardo. Se i risultati si raggiungono, di certo non sono mai merito di una singola persona, ma di un gruppo. Per questo un grazie particolare lo devo a colei che mi ha "indicato la via" gettando le basi di questa ricerca con i suoi precedenti lavori e che ha inseguito con me questa idea, la Ricercatrice Arianna Pipitone. Infine voglio ringraziare tutti i miei colleghi, ma soprattutto amici di laboratorio e...di serate: Vito Gentile, Camilo Duarte Eraso, Fabrizio Milazzo e tutti coloro che hanno contribuito a rendere leggere le mie giornate in laboratorio.

Difficile elencare tutti, ma è proprio a tutti che va il mio grazie per il vostro contributo, il tempo e la pazienza che mi avete riservato.

Palermo, 23/02/2018

Acknowledgments

The work you are about to read is, for you that are reading it, the summary of a research work carried out over a three-year period by a research student. In fact, for the one who wrote it, it represents the concretization of what initially considered an idea, the result of an “ingenious fantasy”, which led him to read, to document and compare himself with the scientific literature, with those who arrived before him and with those who, in the lab and in life, was next to him. In few words, it represents the true meaning of Research: making reality what was not reality, before now.

As you know, in the real life, no peaks are reachable through downhill roads, indeed you will always have to follow bristling and full of pitfalls paths. What gave me the strength to reach this goal was the awareness of never having been alone because at my side I had a woman of those that any man aspires to have next to him and that I have the grace to have, as sweet as strong, my girlfriend, Adriana. She has not stopped even for a moment, together with his brother Attilio and his parents, to encourage me. Together with her, I want to thank my parents, my shields and my armor, who protected me, never failing me to give their support under all points of view. A big family is a great blessing and for this I can only say thanks to my sister Cinzia and Domenico that have contributed in recent years to make it more numerous and happy giving me my

two beautiful granddaughters: Maria Cristina and Sofia, those who have made my days less "theoretical" and more "experimental", during my periods of study at home.

The moments of cultural growth in these three years have been many, but I can not emphasize one in particular, the months spent at the Vrije Universiteit of Brussels where I met tutors and colleagues of which I will carry with me a wonderful memory. Starting from the great Professor Luc Steels, together with Professor Katrien Beuls, Remi Van Trijp and Paul Van Eecke, that have accompanied me in this experience of growth. Together with them, I would like to thank all the colleagues of the AI Lab who have been at my side in this experience, making me feel in Brussels like at home. Thanks: Piera Filippi, Yannick Jadoul, Michael Anslow, Heikki Rasilo, Bill Thompson, Andrea Ravignani, Joan Loeckx and I hope not forgetting other Labsters (in case it is the fault of the lack of beer/warm wine!).

Last, but absolutely not the least, I would like to thank those with whom I have spent more time in these three years than with any other person, starting from my tutors Proff. Edoardo Ardizzone and Roberto Pirrone, men of Science, always available to dialogue, that have accompanied me in this milestone. If the results are achieved, certainly it is never the merit of a single person, but of a group. This is because I wish to give a special thanks to the one who "showed me the way" by laying the foundations of this research with her previous works and who pursued this idea with me, the Researcher Arianna Pipitone. Finally I want to thank all my colleagues, but especially friends in the laboratory and ... evenings: Vito Gentile, Camilo Duarte Eraso, Fabrizio Milazzo and all those who helped to make happier my days in the lab.

It is difficult to list all of them, but it is to all of them that my thanks go

out for your contribution, the time and the patience you have given me.

Palermo, 23/02/2018

Contents

List of Figures

List of Tables

Glossary

CxG	Construction Grammar
FAQ	Frequently Asked Question
FCG	Fluid Construction Grammar
HCI	Human-Computer Interaction
HPSG	Head Driven Phrase Structure Grammar
IR	Information Retrieval
KB	Knowledge Base
MtF	Mapping to Forms
MtM	Mapping to Meanings
NEEL	Named-Entity rEcognition and Linking
NER	Named-Entity Recognition
NL	Natural Language
NLI	Natural Language Interface
NLU	Natural Language Understanding
NLP	Natural Language Processing
OWL	Ontology Web Language
POS	Part of Speech
QA	Question and Answering
QAI	Question and Answering Interface

RDF	Resource Description Framework
RDFS	RDF Schema
SPARQL	SPARQL Protocol And RDF Query Language

Chapter 1

Introduction

1.1 Motivations and Challenges

Part of the Semantic Web perspective is to provide web-scale access to contents that are described semantically. In particular, this implies understanding users' information needs accurately enough to allow for retrieving a precise answer using semantic technologies. Currently, most web search engines are based on purely statistical techniques. While they are not able to figure out the meaning of a query, they can provide answers by returning the statistically most appropriate answer to a user's query-based on some measures for computing similarity in vector space (Baeza-Yates et al., 1999). Information Retrieval (IR) techniques applied to the Web have gained a reasonable degree of maturity, which is clearly corroborated by the success of search engines such as Google, Yahoo and the like. These search engines are providing a baseline quite difficult to outperform. Due to the nature and the maturity of the underlying statistical techniques, they are more robust and scale to the size of the Web, as opposed to semantic technologies. For restricted domains which can be formalized using

Figure 1.1: Semantic web vision.

ontologies, there is nevertheless the hope that semantic technologies can be put into work to allow for more semantics based search. One of the crucial steps within such an endeavor is to precisely capture the user's information need. But how does the user express her/his information need? If we look at the widespread usage of web search engines, we can conclude that users are definitely used to express their information need via simple queries based on keywords. In opposite, using natural language should represent a more efficient and precise way in understanding user's needs than keywords specifying.

1.1.1 Natural language communication

A natural language communication system, being a method to interface humans and machines, faces us to the problem that all the linguistic material could be used in different ways, so the receiving artificial system could be in the condition to not understand the real meaning of the user message if this system is not able to extract enough information from the received content.

What distinguishes natural language understanding from the comprehension of other languages is the need of using a "language knowledge" (Bos, 2011), in a way that it is necessary for the system to know what is the real meaning and role of a word in a text.

Assuming that grammatical structures vary based on the used language, a first problem that has to be dealt with is related to the learning of grammatical forms and, in particular, it was referred to the Cognitive Linguistics (Geeraerts and Cuyckens, 2007) symbolic approaches. These approaches derive grammatical structures from semantic networks (such as ontologies) and dictionaries, rather than from large corpora, written in the same language.

Looking at the state of the art of natural language processing, may production systems have been already devised that allow users to extract information content from the analysis of texts. Basically, the research in the field of language processing, in the six decades ranging from its beginnings until now has mainly seen the development of systems based on standard theories of information processing that can be summarized in the following five main categories (Jurafsky and Martin, 2000) :

- State machines
- Rule systems
- Logic
- Probabilistic models
- Vector-space models

However, in most cases, such systems are limited to relate the criteria and requirements specified by the user, with the text being analyzed. Most of the systems currently manufactured and available in the literature, which pose similar goals, have some limitations, first of all extracting the semantic content of an entity present within a text, that is the single word, while estranging it from the surrounding context. This approach carries obvious problems of semantic ambiguity. Consider the case of a text in which a date is present. Going to consider the single entity one can understand that this is a date, and extract precise temporal information but it can not be established what the date refers to i.e. a birth date, and who was born on that date or if the date refers to the creation of a paper, and what paper it is. If, by contrast, one analyzes the context adjacent to it, or the whole structure of the sentence in which it is reported, more information can be inferred (Lakoff, 1987).

Another problem arises when an entity is expressed in different terms and ways than that the agent has in mind in relation to that entity. Such a case could arise in the use of a synonym, or, in a more complex way, using a metaphor.

As a result, these studies have led to considerable results regarding the implementation of systems such as search-engines, while showing that the study of natural language understanding is, again, an issue that leaves ample room for debate (Bos, 2011).

From these brief considerations it can be understood how computational semantics turns out to be of considerable interest among the many research areas involved in natural language processing, given the need of being able to capture the precise meaning of expressions in natural language in order to allow an artificial agent making the correct inferences both in speaking and comprehension tasks.

1.1.2 The informal language and its linguistic issues

The main characteristics of informal language are ill-formed syntax (Han and Baldwin, 2011) and dynamicity (Kobus et al., 2008); ill-formedness is related to the presence of abbreviations, unconventional spellings, and unstructured expressions that give to the text a noisy lexical nature. Dynamicity regards the fact that informal language changes more frequently than the formal one; there are not strict grammatical rules for describing the informal language, however the sentences expressed in such a language can be understood by a reader or a hearer even if they do not have rigid structures. Moreover, the informal language is characterized by the possibility to add new grammatical units or delete old ones over time: either new structures can emerge during conversation or existing ones can be removed if they are not frequently used.

Substantially, social web users contribute to rapid emergence of new grammatical forms, symbols, and unconventional expressions as they own the skills to understand and re-use such novelties. This is one of the reasons for which the traditional Natural Language Processing (NLP) techniques fail when applied directly to informal texts. In particular, statistical approaches do not perceive such linguistic changes until huge annotated text corpora are at disposal for the scientific community; manual intervention through explicit annotations of data is needed in this perspective to allow even very good performances for these techniques. Challenges are a way to create such reference corpora, and this is an activity that has to be repeated frequently to address changes in language along with re-training and fine tuning of such systems.

1.2 Goals and Solutions

Starting from the semantic web vision in which open-data and rich ontology-based semantic markup are becoming widely available, new possibilities of interfacing natural language and ontologies have been investigated, using the symbolic principles and theories of computational semantics. By the analysis of all the systems and technologies presented recently in the scientific literature, different problems arise that need a suitable solution, and that are the main object of this research.

The main contributions of the work presented in this dissertation are aimed at attempting to solve such problems. An architecture was defined that makes use of Fluid Construction Grammar (FCG) (Steels and De Beule, 2006) whose goal is facing the problem of linking ontology concepts and their correspondent natural language expression. In particular, relying on Construction Grammar formalism, which will be explained further, an architecture component has been developed that links ontology concepts to the semantic pole of constructions in FCG, while natural language expressions are linked to the syntactic one. The concept of "fluidity" of constructions makes the system able to annotate text with a relevant degree of tolerance to grammatical errors or omissions, which are typical of informal text.

Semantic disambiguation has been faced through the definition of a suitable "similarity measure" natural language expressions under investigation and the potential corresponding concepts in the ontology. This measure is given by the analysis of several features of the expression. Both the syntactic and semantic roles of the expression in the utterance have been considered to build the measure, while statistical information have been used to assess if the expression can be related with ontology concepts near the strictly eligible ones. Also the information coming from taxonomies and thesauri like WordNet (Fellbaum, 1998a) and FrameNet (Baker et al., 1998) has been used. WordNet was used to obtain word synonyms and synsets, while Framenet was intended for gaining possible argument structures for verbs.

The proposed architecture was employed for Question Answering (QA) and to solve specific NEEL task in informal text, in particular implementing a system for tweet annotation. The aim of this contribution is avoiding statistical analysis, considering that social information has a fast and rapid changing over time, so an attempt has been carried out to perform an analysis that is not influenced from previous trends and collections.

Named Entity rEcognition and Linking (NEEL) is a sub-task of information extraction that aims at locating and classifying each named entity mention in text into the classes of a knowledge base. The interest for such a task has been growing exponentially with the advent of the Web 2.0 technologies, leading to the Social Semantic Web research field; unlike the Semantic Web that is considered a model for solving the epistemic interoperability issues, the Social Semantic Web makes users free to publish uncontrolled texts without grammatical constraints, spreading them to a multitude of other people. Semantic annotation of social data by linking them into structured knowledge attempts to control such phenomena, making the social data both machine-readable and traceable.

The proposed strategy is inherently unsupervised to reach the goal of being sensitive to both dynamicity and ill-formed syntax, while avoiding manual annotations: even if we make some linguistic considerations about the tweet, they are not based on strict grammatical rules, which also need frequent manual updating to cope with language dynamicity. Rather, the rules implemented here attempt to simulate the cognitive processes that can be involved in understanding the meaning of a tweet, and produce semantic annotation as their outcome. The similarity measure and the cognitive processing rules are properly re-defined for highlighting the particular kind of both the text, that is a tweet with its typical features, and the task. In a few words, the main focus is on the definition of new aspects in the implementation of the cognitive processes to allow linking a tweet's informal text to the ontology describing the knowledge domain. DBPedia has been used in this respect. Such processes consider the particular structure of a tweet (mentions, hashtags, and partially structured statements) and the nature of the NEEL task. The whole approach is based also on linguistic considerations about the informal language.

1.3 Logbook of Contributes

This section illustrates the working steps done to obtain understanding and verbalization of natural language sentences related to a knowledge domain that is modeled using ontology; the main aim is the creation of an open-domain QA System.

In the first attempt, a simple agent able to process the user's query was developed in order to understand the query topic, and the particular features about the subject requested by the user. To achieve this target the attention was focused on interfacing natural language with ontology. The very first version of the agent used simply POS tagging to make a syntactic analysis of the input query and producing an ontology query to retrieve the required information through the ontology-FCG connection.

Going on, the agent was improved through the implementation of a recovery strategy for those queries such that no good match related to the query topic could be recovered in the ontology properties. In this case, the agents starts analyzing possible non structured text coming from different sources, and related to the query subject. In this case, the use of textual similarity measures was introduced and some tests were carried out to achieve good parameter tuning.

The introduction of support text analysis and similarity measures added a serious improvement and the good experimental results bring the system to be published in (Pipitone et al., 2016c).

The next attempt was aimed at improving the linguistic knowledge of the agent, especially in the understanding phase. The σ-expander module was implemented, which expands the language vocabulary of the agent using online dictionaries. In the effort to create a multi-language agent, the Wiktionary resource was interfaced, and several tests were carried out both in English and Italian.

This version of the agent was used to solve the QA4FAQ task at EVALITA 2016 (Caputo et al., 2016) with more than gratifying results.ChiLab4It was the best score application among the participating teams, and the only one which outperformed the baseline score proposed by the task authors.

Such encouraging results were the premise for more improvements that were oriented to entity detection in text, so the effort was concentrated in the specific task of both recognizing and disambiguating entities through the analysis of phrase structure. The computational formalism implemented by Fluid Construction Grammar was the enabling technology, which allowed considering the information content conveyed by the phrase syntactic features to the specific detected entity. This approach allows the agent to achieve its goal not only in grammatically correct text, but also in texts with loose syntax, such as informal text written in SMS and Twitter messages.

This latest version of the agent was tested for English language using the #Micropost2016 dataset, and performed very well compared with other state of the art supervised and unsupervised approaches. This version of the agent performs the first best score among the systems that adopts unsupervised approaches and reaches the absolute second rank of the whole competition.

In view of the previous considerations, the result of the present research can be regarded as a novel state-of-the-art multi-language QA system which implements a cognitive architecture for understanding and producing utterances in natural language.

1.4 Dissertation Outline

A description of the further chapters of this dissertation is here reported, in particular this work is organized as follows.

In chapter 2 a survey of ontology-based NLP systems reports the current state of the art, introducing the reader to the various kind of tools proposed by the current scientific literature. The chapter goes on describing the main differences, pros and cons of the statistical and symbolical approaches, the two main approaches on which the QA systems are based. The study of the state of the art continues with a survey of the main NLP software specifically entailed for informal language processing, highlighting their main features and weakness. Subsequently it will be introduced the Construction Grammar formalism and its computational implementation, Fluid Construction Grammar. It will be explained the main principles of the formalism and the Unification and Merging mechanism which is the fundamental principle on which this formalism is based.

At the end of the chapter, in the light of what has been seen, the main detected issues to investigate will be exposed and the proposed solutions on which this research activity has head on.

In chapter 3 the model proposed to overcome the issues investigated, the QuASIt cognitive architecture, will be introduced. In particular it will be explained how, in the proposed model, the interpretation and/or production of a natural language sentence requires the execution of some cognitive processes over both a perceptually grounded model of the world, and a previously acquired linguistic knowledge. In particular, two kinds of processes have been devised, that are the *conceptualization of meaning* and the *the conceptualization of form*, that will be explained in depth.

In chapter 4 a series of applications are described, whose implementation was inspired according to the QuASIt cognitive model. Specifically, ChiLab4It is an evolution of the QuASIt architecture specialized to address the particular task of selecting the most relevant FAQ among those contained in a given *FAQ base* according to the question asked by the user. In the second study case, was developed a system able to use the QuASIt cognitive model to solve the NEEL task on informal text sources.

Chapter 5 reports the experiments and the datasets adopted to test and evaluate the architecture here proposed. It will be described the tasks in which the various developed applications were employed and the outcomes are compared with the others state of the art tools.

The final chapter will discuss about the outcomes of this research activity, the main strengths of the proposed model and the open issues in which can be projected the future efforts.

1.5 Publications

Parts of the work in this thesis have been published in several referred conference proceedings:

- Pipitone, A., Tirone, G., Pirrone, R. (2016). QuASIt: a cognitive inspired approach to question answering for the Italian language. *In AI*IA 2016 Advances in Artificial Intelligence (pp. 464-476). Springer International Publishing.*
- Pipitone, A., Tirone, G., Pirrone, R. (2016). ChiLab4It System in the QA4FAQ Competition. *In CLiC-it/EVALITA*.
- Pipitone, A., Tirone, G., Pirrone, R. (2017). Named Entity Recognition and Linking in Tweets Based on Linguistic Similarity. *In AI*IA 2017 Advances in Artificial Intelligence (pp. 101-113). Springer International Publishing.*

Chapter 2

State of the Art

2.1 Ontology-based natural language processing systems

Ontologies have been designed to capture the semantic knowledge of a domain in a machine understandable form. Current standards for ontologies managing, like OWL, are lacking in linguistic grounding, and are not able to achieve a clear link with natural language.

With billions of triples being published in recent years, such as those from Linked Open Data, there is a need for more user-friendly interfaces which will bring the advantages of these data closer to the casual users. Research has been very active in developing various interfaces for accessing structured knowledge, from faceted search, where knowledge is grouped and represented through taxonomies, to menu-guided and form-based interfaces such as those offered by KIM (Popov et al., 2003). While hiding the complexity of underlying query languages such as SPARQL2, these interfaces still require that the user is familiarised with the queried knowledge structure. However, casual users should be able to access the data despite their queries not matching exactly the queried data structures (Hurtado et al., 2009). Natural Language Interfaces (NLIs), which are often referred as closed-domain Question Answering (QA) systems, have a very important role as they are intuitive for the end users and preferred to keyword-based, menu-based or graphical interfaces (Kaufmann and Bernstein, 2007).

Most QA systems contain the classifier module which is used to detect the question category or the type of the question. The successful parsing is based on this identification. However, the syntactic patterns for this classification are usually derived from the dataset which must be large in order to work efficiently. Moreover, as Ferret et al. point out: "answers to [some] questions can hardly be reduced to a pattern." (Ferret et al., 2001). In addition, it is not trivial to translate successfully parsed question into the relevant logical representation or a formal query which will lead to the correct answer (Tang and Mooney, 2001).

Bridging this gap, unskilled users could be able to infer the information described in the ontology and it would be possible either producing or parsing utterances about the represented domain automatically.

Natural Language Processing (NLP) in combination with ontologies received a lot of attention recently. The ontology based understanding of NL and the translation into SPARQL combines the research of several recent publications. (Lakoff, 1987) propose an approach to ontology-based interpretation of keywords for semantic search. Keywords are mapped to concepts in the ontology, and the graph then is further explored to retrieve available relations between these concepts. These relations most likely are gaps in the query, which are not specified by the user. The user knows these gaps intuitively, e.g. searching for

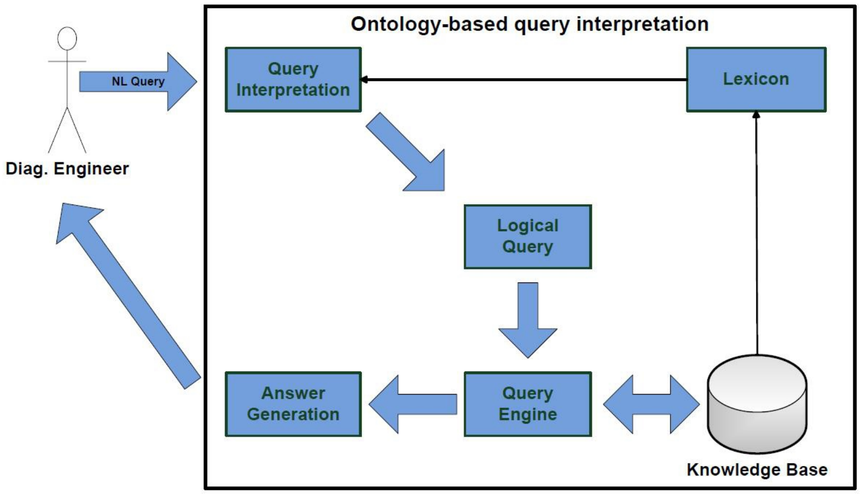

Figure 2.1: A general approach to ontology-based query interpretation.

an author name and a book title, the obvious relation would be "authorOf" or the opposite "writtenBy". To avoid exponential search time or dead locks, they define an exploration width to limit the amount of visited nodes.

Another possibility to translate NL to SPARQL has been carried out with AutoSPARQL (Lehmann and Bühmann, 2011). They use active supervised machine learning to generate SPARQL queries based on positive examples. Starting with a question and following up with answering, estimating whether an entity belongs to the result set, will continuously improve the algorithm and the results. This approach leads to very good results. One problem is the portability to a different Knowledge Base (KB). Depending on the size, the effort of learning the positive and negative examples will increase drastically with the size of the KB. SPARK (Zhou et al., 2007) is a prototype for processing NL keyword requests. The output is a ranked list of SPARQL queries. The major process steps are: term mapping, query graph construction and query ranking. The query ranking is a probabilistic model based on the Bayesian Theorem (Smets, 2008) . The authors claim "encouraging" translation results.

The problem here is, that choosing an option out of the ranked query list requires expertise in SPARQL and the underlying KB.

QuestIO (Damljanovic et al., 2008) is a Question and Answering Interface (QAI) to query ontologies using unconstrained language-based queries. It requires no user training and the integration in other systems is simple. During processing, every single human understandable phrase is extracted from the ontology (classes, properties, instances, relations). Relations are ranked with a similarity score which takes into account the name, the position and distance in the ontology hierarchy. The advantages of this approach is the simple structure, which allows queries with random length and form as well as the slight effort for customization. A problem is the lack of basic NLP operations. No word stemming is used, which conflicts with the statement of "queries with random form", since synonyms or different grammatical forms of a word may lead to no or even wrong results.

FREyA (Damljanovic et al., 2011) is a successor tool of QuestIO to interactively query linked data using natural language. In contrast to QuestIO, FREyA utilizes syntactical parsing combined with an ontology based lookup to interpret a natural language question. If the intention behind the query remains unclear, the system involves the user with a QAI. The answers of the user are used to train the system and improve its performance and accuracy over time. Depending on the knowledge of the user, answering questions of a specialized domain model and vocabulary is difficult or impossible (Unger et al., 2012).

NLPReduce (Kaufmann et al., 2007) is another "naive" approach to introduce a Natural Language Interface (NLI) to query a KB in the semantic web. The natural language processing is limited to stemming and synonym expansion. NLPReduce does not claim to be an intelligent algorithm. It maps domainindependent words and synonyms to their expressions used in the KB.

The work proposed by (Estival et al., 2004) carries out research for ontology based approaches to process NL. They state two options for the facilitation of NLP with the assistance of ontologies. Firstly, using an ontology to build a lexicon and defining terms (concepts and relations). Secondly, an ontology represents a KB in a formal language and provides further knowledge to conduct more complex language processing. Both insights are valuable with regard. Building up a lexicon and integrating additional knowledge, that is not represented in the base data, supports the formulation of search queries from NL input.

2.2 Statistical vs. Symbolical approaches in QA systems

Human natural languages are open systems; they exhibit a high degree of evolution in short time. Evolution depends on speakers, which tend to change expressiveness in each situation of real life, rapidly breaking the linguistics conventions. When artificial systems (like QA systems) interact with humans, these variations represent a significant problem; often QA systems fail because their linguistic model is not able to deal with either new meanings or new expressions emerging during a single dialogue session.

In statistical approaches (all the tools presented to the well-known competition TREC LiveQA (Dean-Hall et al., 2015) and those described in (Boubiche et al., 2015)), such changes might be not sensibly observable despite frequent training, and the QA system might not adapt to new sentences or catch users' attention with correct interactions. Other approaches try very hard to separate the issues concerned to efficiency from issues concerned to grammar representation (Sag et al., 2003). Also, if the linguistic sources are not exhaustive for a language, such approaches might not be used; this challenge was taken up in (Basili et al., 2004) where a new ontology-based QA system is defined. Such a system extracts data from a federation of websites, developing a multilingual environment, which implies the ability to manage several languages and conceptualizations. However, in this approach a large use of linguistic sources is made, because they are linked to the domain ontology by (Basili et al., 2003).

Undeniably, the underlying computational linguistic model of an artificial agent should handle the *fluidity* of language to face the problem outlined above. The model shown in this dissertation tries to do this inspired by both the *Constructions Grammar* (CxG) (Hoffmann and Trousdale, 2013) and the cognitive processes, which are the basis of procedural semantics (Spranger et al., 2012). CxG is a "symbolic grammar" because all elements have a surface form that is the symbolic representation of the element in the human's mind. Grammatical structures have symbolic representations too, which are the conjunctions of elementary items. All these elements (both structures and items) are considered as tied and related intrinsically to other knowledge structures in the mind. The basic units of CxG are the *constructions*; a construction is a regular pattern which has a conventionalized meaning and function (Goldberg and Suttle, 2010). The meaning side of a construction is captured in a *semantic pole*, while all the aspects related to form, as the syntax, are captured in a *syntactic pole*.

In (Hoffmann et al., 2013) a psychologically plausible account of language was made, by investigating some general cognitive principles. Differently from the approach presented here, authors keep into account the linguistic problems only. Instead, we want to integrate both linguistic and world knowledge, according to cognitive linguistics mainstream.

With the aim of generalizing our linguistic model, the main properties we referred to are the *continuum* and the *abstract categorizations* of constructions. The continuum is the result of the taxonomy of constructions on which the grammar relies on; quite general constructions subsume the so-called *item-based* ones, that are built out of lexical materials and frozen syntactic patterns, according to such a taxonomy. The continuum is realized through the constructions' open slots where sentences with specific semantic and syntactic structures can fit. The semantic and syntactic categorizations are the means by which constructions relate meaning to form, and allow the *conceptualization of meaning*. As an example, many languages categorize the specific roles of the participants in an event represented by the verb in terms of abstract semantic categories like agent, patient and so on, before mapping them into abstract syntactic categories like the nominative case. Syntactic categories translate further into surface forms.

Categorizations allowed us to abstract the linguistic typology of a query, which is next fitted to the user questions. Such an approach is obviously much more efficient than having an idiosyncratic way to express each question because fewer constructions are needed, and new queries can be understood even if the meaning of the whole question is unknown; this represents also the solution of the fluidity issue.

The only existent computational version of CxG is the Fluid Construction Grammar (FCG) (Steels, 2011a) that is an engine that implements both parsing and production using the same set of constructions. FCG is based on two mechanisms: *unification* and *merging*.

In parsing, a *transient structure* owning only the syntactic pole is fitted with the set of constructions; when a construction is unified, the transient structure is merged with its slots, and new slots are added in the semantic pole.

Production uses the same mechanism by swapping semantic and syntactic pole as the initiator of the process; the transient structure starts owning only the semantic pole that is unified with constructions, and the syntactic one is next merged.

2.3 NLP for Informal Language

Due to their inherent nature, tweets are noisy and short so the performance of standard NLP software significantly suffers. Derczynski et al. (Derczynski et al., 2015) demonstrated that the performance of various state-of-the-art NLP software (e.g., Stanford NER and ANNIE) is typically lower than 50% F_1 for tweets.

Many studies, such as (Beaufort et al., 2010) and (Kaufmann and Kalita, 2010) perform "informal language normalization" for disambiguating informal tokens; normalization is achieved defining a set of correspondences with the traditional natural language, that are called the *formal counterparts*. In particular, (Liu et al.) employs (external) web mining to collect the counterparts of informal tokens. To optimize the labeling process, a large set of tokens are put automatically in correspondence with words by searching for the informal tokens in Google, and selecting the counterparts from the top 32 snippets based on the length of the shared character sequence.

In (Habib and van Keulen, 2015) the authors propose a strategy for entities disambiguation that refines their previous works (Habib and van Keulen, 2013). They start with an initial extraction-like phase aimed at finding mention candidates, based on segmentation and KB lookup. Next, the disambiguation process is applied to the extracted candidates; this process uses three typical modules (the matcher, the feature extractor, and the Support Vector Machine ranker) and gives extra features to the mention candidates. Finally, another classification extracts mention candidates as either true or false entities, using the features obtained from the disambiguation process. Authors in (Ritter et al., 2011) integrated some typical NLP processes for extracting entities from informal tokens, while in (Li et al., 2012) authors attempt to discover a global context of informal sentences from both Wikipedia and Web N-Gram corpus. The sentences are segmented by a dynamic program, and each of these segments is a candidate entity. Finally they rank segments according to their probability of being an entity. In (Wang et al., 2012) a linguistic analysis is carried out to make prevision on tweets; differently from the approach proposed here, the authors adopt a statistical approach that estimates trends and distributions into a big collection of tweets.

An important set of systems is referenced in (Rizzo et al., 2016), that is the final report of the #Micropost2016 workshop NEEL Challenge co-located with the World Wide Web conference 2016 (WWW '16) which was used in the experiments carried out for evaluating the approach presented in this dissertation.

2.4 Construction Grammar and Fluidity

Human languages are inferential communication systems gives them a number of special properties. Between them, main property is that languages can be openended: at any moment the set of available conceptualizations and linguistic conventions can be expanded by speakers if they need to express something that was not yet conventionally expressible in the language, because hearers are assumed to be intelligent enough to figure out what was meant and possibly adopt any innovations introduced by speakers.

This *fluid* character of human language helps to make them adaptive to the needs of language users that keep changing as human societies evolve and become more complex. One of the explicit goals of Fluid Construction Grammar is to try and deal with fluidity of the languages, confering Construction Grammar this property.

This topic is discussed more extensively in (Steels, 2011b).

2.4.1 Construction Grammar

Construction Grammar (CxG) is an approach for studying linguistic structure first proposed in (Fillmore, 1985), (Fillmore et al., 1988), (Lakoff, 1990), that shares certain assumptions with both formal liguistic theories as (Jackendoff, 1990) and the cognitive ones, that is the study of the language in its *cognitive function*, where cognitive refers to the crucial role of intermediate informational structures in our encounters with the world.

Construction grammar theories consider *constructions* as the basic units of language. While what makes up a construction has been different for the different theories of construction grammar, it is generally approved that a construction is a *"syntactic pattern which is assigned one or more conventional functions in a language, together with whatever is linguistically conventionalized about its contribution to the meaning or use of the structures containing it"* [1]. Constructions can be adapted easily to changing language patterns: they consider both semantics and syntax of the lexicon, and are easier to manage than words as the atomic unit. For this reason, constructions can be semantically computed and this allows their integration into bigger collections.

[1] see Fillmore et al. (1988), pag. 36.

Different construction definitions can only be posited if there is something about the form or meaning of the construction itself that is not given from ordinary compositional processes, from the literal meaning, from the processes of conversational reasoning, or from other constructions that exist in the language (Kay and Fillmore, 1999).

What makes a construction a construction is that it possesses *"properties of form (syntactic and phonological) and meaning (semantic and pragmatic)"* (Croft and Cruse, 2004).

Linguistic Requirements

The linguistic perspective of CxG is in the general line of cognitive grammar (Langacker, 1999) and more specifically construction grammar (Goldberg, 1995). This means the following (Steels and Beule, 2006):

- *CxG is usage-based.* It means that words available to speakers and hearers consist of patterns which can be highly specialized, perhaps pertaining to a single case, or much more abstract, covering a wide range of events. New sentences are constructed or parsed by assembling these patterns through the unify and merge operators.

- *The grammar and lexicon are modeled by symbolic units.* A symbolic unit associates aspects of semantics with aspects of syntax. They feature a semantic pole and a syntactic pole, can be bi-directional, so they can be used for both production and parsing (as in the case of Fluid Construction Grammar). Uni-directional constructions may exist such as in Embodied Construction Grammar.

- *There is a continuum between grammar and the lexicon.* Not only templates can be at different levels of abstraction, but there is also no formal difference between the structures of lexical and grammatical entries. In the case of lexical entries, the syntactic pole tends to be a lexical stem and the semantic pole covers some concrete predicate-argument structure.

  In the case of grammatical constructions, the syntactic pole contains various syntactic categories that constrain the sentence, and the semantic pole is based on semantic categories, otherwise there is no formal difference between the two types of templates.

- *Syntagmatic and Paradigmatic Compositionality.* To produce or parse a phrase, templates can be combined (several templates all matching with different parts of the meaning in production or with parts of the sentence in parsing are simply applied together) or integrated (using hierarchical templates that combine partial structures into larger ones). A part from this syntagmatic composition, there is also a paradigmatic compositionality, that means the possibility that several templates are covered and each contribute additional constraints to the final sentence. Both forms of compositionality are completely supported with the unify and merge operators defined later.

- *Schematization occurs through variables and categorization.* A template has the same form as an association between a semantic structure and a syntactic structure, in other words both poles of a template are *feature structures.* However, templates are more abstract (or schematic) in because variables are used instead of units and values, and syntactic or semantic categories are introduced to restrict the possible values of the semantic and syntactic pole. These categories are often established by syntactic or semantic categorization rules.

2.4.2 Fluid Construction Grammar

In recent years, only very few demonstrations of non-trivial grammars are considered in grounded artificial production and understanding of natural language utterances. Part of the problem of grammar emergence for artificial verbalization is that a grammar is more encompassed than lexicon. Studies and experiments on grammars require powerful techniques from symbolic processing; existent formalisms are not strongly linked to one or the other linguistic theory, as happened for example for the Head Driven Phrase Structure Grammar (HPSG) (Pollard and Sag, 1994).

HPSG is a kind of dependency grammar and it is centered around the *head* of a phrase that is linked through the *head-dependent* relations to other roles of the words in the same phrase. Roles can be among *modifiers*, *specifiers* and *complements*. Although this theory is very useful and computing roles is simple, not all the linguistic phenomenons can be defined in terms of them, and there is no general consensus for these settings. The primary goal of HPSG is to build a theory of the knowledge embodied in the human brain, and not to build agents that use this knowledge (Pollard, 1997).

Most other formalisms basically find a minimal but necessary set of grammatical rules and principles such that *empirical linguistic data* satisfy the grammar: questions of how and why such linguistic data could be produced, learned and evolve are not considered. These formalisms control semantic and syntactic categories for their purposes, closing them.

In order to overcome these limitations, the Artificial Intelligence Lab of the

University of Brussels in the person of Luc Steels together with the researchers at Sony CSL in Paris have been developing for many years a formalism that can handle both production and parsing, and that would be adequate to study natural language grammars: the result of these efforts was the design of a framework named Fluid Construction Grammar (FCG).

FCG is the first available operationalization of Construction Grammar that uses many existing and widely accepted notions in theoretical and computational linguistics, as the *feature structures* that represent both syntactic and semantic information during parsing and production, and *abstract templates* for the representation of lexical and grammatical usage patterns, as in (Sag et al., 1999) or (Bergen and Chang, 2003). These properties make the framework appropriate for our experiments, because we build constructions from the semantics devised from the ontology, and couple semantics with syntax through a suitable set of rules. Allowing both production and parsing through the same constructions, we can verbalize about the ontology, and understand free text related to the knowledge domain it describes.

FCG is based on the general operations of *unification* and *merging*. Differently from other formalisms, FCG attempts to investigate the origins and the evolution of semantic and syntactic categories considering them free in principle. FCG is more concerned with things like flexibility, learning, invention, usage and other creative aspects of language. Making the categories free is one of the main features of the whole framework.

FCG Linguistic Feautures

Moreover, open categories are in line with the Radical Construction Grammar approach which argues that linguistic categories are not universal and subject

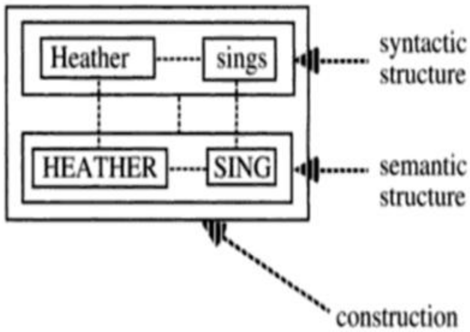

Figure 2.2: A construction with its semantic and syntactic poles.

to evolution (Croft, 1991).

Feature Structures

As mentioned before, unit structures hold the information about the utterance to process. They are represented as a list of units. For instance, let consider the sentence "Paul hates Janet".

As a first step it is necessary to build a unit for each word in the sentence, and one additional super-unit called Top-unit to keep the other three units together. Units are represented as *lisp like lists*, see e.g. (Steele, 1990), for which the entire list is delineated by plain brackets as:

```
(Janet-unit (form ((String "Janet"))))
```

The above expression is a unit. It is a list where first element is the *unit's name*, in the example the symbol Janet-unit, and second element is a list of type `(form ((String "Janet")))`, which is the only `feature` of this unit. The feature's name again is the symbol `form` and its value is the list `((String "Janet"))` and so on.

Generally, units will be represented as lists, with the first element that represents the unit's name and all remaining elements its features. Features will also be represented as lists, again with the first element as its name and a value as second element. The name must always remain the first element in the list; unit structures will be lists of units. Hence, in the example, there is a unit structure that contains a unit for each word in the phrase "Paul hates Janet" and one additional super-unit (called *Top-unit*) to keep together the other three units. They look like:

```
((Top-unit (syn-subunits (Paul-unit Janet-unit Hate-unit))
(form ((meets Paul-unit Hate-unit)
(meets Hate-unit Janet-unit))))
(Paul-unit (form ((string "Paul"))))
(Janet-unit (form ((string "Janet"))))
(Hate-unit (form ((string "hates")
(stem "hate")))
(syn-cat ((lex-cat verb)
(number sing)
(person 3rd)))))
```

Many other linguistic formalisms (e.g. hpsg and ecg) represent feature structures with a boxed notation or as attribute value matrices instead of with the bracketed lisp-like notation shown here. In such a notation the above unit structure for the sentence "Paul hates Janet" could be represented as:

$$
\begin{bmatrix}
\text{Top-unit} & \\
\text{FORM} & \left\langle \text{meets}(\boxed{1},\boxed{2}),\ \text{meets}(\boxed{2},\boxed{3}) \right\rangle \\
\text{SYN-SUBUNITS} & \left\langle \begin{array}{l}
\boxed{1}\begin{bmatrix}\text{Paul-unit} & \\ \text{FORM} & \langle\text{STRING "Paul"}\rangle\end{bmatrix} \\
\boxed{2}\begin{bmatrix}\text{Hate-unit} & \\ \text{FORM} & \langle\text{STRING "hates"}\rangle\end{bmatrix} \\
\boxed{3}\begin{bmatrix}\text{Janet-unit} & \\ \text{FORM} & \langle\text{STRING "Janet"}\rangle\end{bmatrix}
\end{array} \right\rangle
\end{bmatrix}
$$

In this notation lists are typically delineated with hooked brackets (like <this>.) Both representations are more or less similar. Whatever the notation used, a unit structure can easily be extended.

In FCG, semantic and syntactic information are kept in different unit structures. Syntactic units normally contain the features *syn-subunits*, *form* and *syn-cat*. Semantic units typically have the features *sem-subunits*, *meaning* and *sem-cat*. The sem-cat feature describes information about the semantic category of the unit, as for example if it is an object or a person. The fact that semantic and syntactic information is kept in different structures reflects that constructions in language are meaning-form mappings. So a lexical construction for "Paul" is a mapping between a syntactic pattern selecting for the string "Paul" in the form feature of a syntactic unit, and a semantic pattern introducing the predicate 'Paul (?x)' in the meaning feature of the correspondent semantic unit.

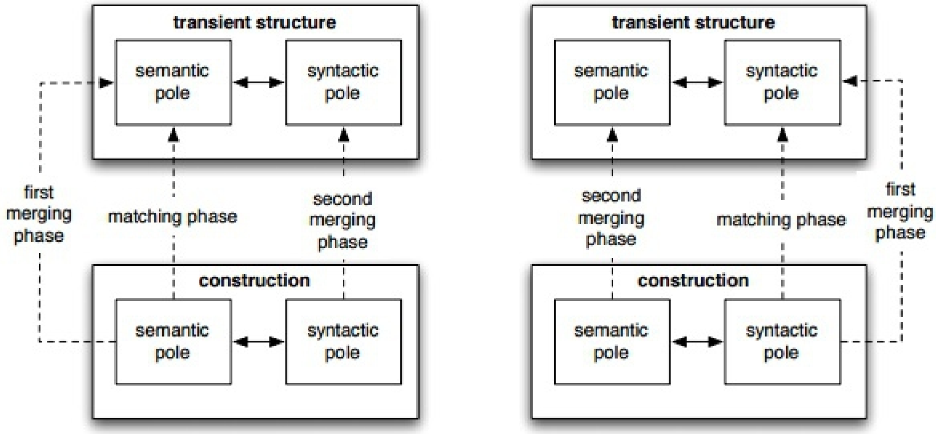

Figure 2.3: Use of a transient structure. Left: production. Right: parsing.

Unification and Merging

Another important aspect of the FCG is the *reversibility* of the grammar; the same set of constructions is used for both production and parsing. This property is realized by a mechanism of unification and merging, that makes the so-called *transient structure*.

A transient structure (a graphic example is in Figure 2.3) represents either the sentence to be parsed or the meaning to produced. In production, units of the semantic pole of a construction are matched against a transient structure before additional semantic and syntactic pole are merged with the structure. In parsing, the same strategy is applied but in the inverse direction.

The operation that decides whether a template matches a specific unit structure is called *unification* between the template and the unit structure. This operation allows that the unit structure contains more units than specified by the template. The result of unifying a template with a particular unit structure is a set of sets of bindings of variables to actual values. For example, the unification of the template:

```
((?unit (form ((string "Paul")))))
```

with the unit structure

```
((Paul-unit (form ((string "Paul"))))
```

is a set containing one set of bindings:

[?unit/Paul-unit].

This can also be represented as:

((?unit . Paul-unit)).

A set of bindings specifies how to make a structure from a template: by substituting the variables in the template by the values they are bound too. It also allows a particular unit in the template to contain more features than specified. The operation is also insensitive to the order of the units in the structure or of the features in a unit. Considering structure templates, we can proceed with specifying constructions as mappings between a semantic and a syntactic template as in:

```
((?unit (meaning ((Paul ?x)))))
<-->
((?unit (form ((string "Paul")))))
```

Generally, the semantic pole is written above the double arrow and the syntactic pole below it. The unification of the syntactic pole with a unit structure results in the set of bindings and of substitution of these bindings in the semantic pole. The operation that takes a unit structure and forms a new extended structure as specified by a template is called merging.

While parsing an utterance, the right (syntactic) pole of a construction is unified with the syntactic structure to see whether it applies. If it does, the left (semantic pole) is merged with the initially empty semantic structure to yield a new semantic structure. While producing a sentence the semantic pole is unified with the semantic structure and, if successful, the syntactic pole is then merged with the initially empty syntactic structure to yield a new syntactic structure.

Summarily, with FCG the information about an utterance are represented with unit structures. Because it is always possible to add units to a structure, or features to a unit or values to a feature, this representation is powerful. We have also set first steps towards representing rules of language as bi-directional mappings between structure templates. These mappings can be used for both parsing and production.

Chapter 3

The QuASIt Cognitive Architecture

3.1 The QuASIt System

The main characteristic of QuASIt (Pipitone et al., 2016c) is the underlying *cognitive* architecture, which stems from the claim that the interpretation and/or production of a sentence in natural language requires running some cognitive processes that use both a perceptually grounded model of the world (that is an ontology), and a linguistic knowledge acquired previously. In particular, two kinds of processes have been devised, that are the *conceptualization of meaning* and the *the conceptualization of form*.

The conceptualization of meaning allows to associating a sense to perceived forms, that are the words of the user query. A sense is the set of concepts of the ontology that explains the form; such a process is implemented considering the ontology nodes whose labels match best the forms from a syntactic point of view. The set of such nodes is the candidate sub-ontology to contain the answer to be produced. The syntactic match is based on a suitable similarity measure.

The second process associates a syntactic expression to a meaning; it implements the strategies for producing the correct form of the answer, once it has been inferred. The form depends on the way QuASIt can be used, that is in both multiple choice and essay questions. In the case of multiple choice questions, the form must be one of the proposed answers. The system infers the correct answer among the proposed ones using the values of the properties' ranges in the sub-ontology; the answer that best matches such ranges syntactically is considered the correct one. If no answer can be inferred in this way, a support text can be used if available.

The support text can be either derived automatically by the system, using the plain text associated to the nodes of the sub-ontology (such as an "abstract" node in the DBPedia ontology[1]) or provided directly as part of the questions as in the case of a text comprehension task.

3.2 The Proposed Model

The proposed cognitive architecture is depicted in figure 3.1. The domain ontology and the CxG represent the knowledge of the world and the linguistic knowledge respectively. We referred to the semantic and syntactic categorizations for defining the cognitive processes of the agent. Two kinds of processes have been devised: the first is related to the *conceptualization of meaning* that associates a perceived external entity (i.e. a word, a visual percept, and so on) to an internal concept. The conceptualization of meaning allows to associate

[1] `http://it.dbpedia.org/`

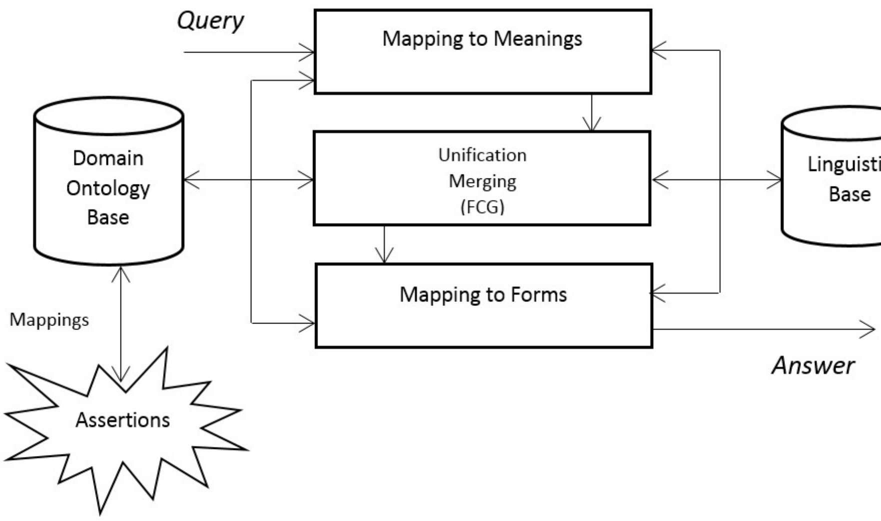

Figure 3.1: The QuASIt cognitive architecture

a sense to a perceived form; in our case, the forms are the words of the query, and the internal concepts are the nodes of the domain knowledge. The second process is related to the *conceptualization of form*, that associates a form or a syntactic expression to a meaning; it is the well-known *lexical access* process. In our case the lexical access is implemented by the strategies for producing the correct form of the answer; such a form depends on the way QuASIt can be used, that is in both *multiple choice questions* and *essay questions*. Generally, the form is the value of a property's range in the conceptualized nodes that are inferred by the system.

For example, in the specific case of multiple choice questions the form must be one of the proposed answers, that is inferred in the same way of the essay questions (i.e. using the values of the ranges of the involved properties). If no answer can be inferred in this way, a support text is used, which is derived from both the text associated to the nodes, such as an abstract, and the text associated to the questions, if available.

In the figure, the knowledge about the world and the linguistic knowledge are located respectively in the *Domain Ontology Base* and the *Linguistic Base*. The *Mapping to Meanings* (MtM) and the *Mapping to Forms* (MtF) modules are the components that model the cognitive processes related to the conceptualization of meaning and form respectively. The *Unification Merging* module is essentially the FCG engine used to perform query comprehension. All the components are detailed in the following sections.

3.3 Domain Knowledge

The ontology forms the structural backbone of the domain, and it represents the terminological box on which the assertions are mapped; assertions are the facts of the domain, and they can be derived from text as in the case of the Wikipedia pages that are mapped to the DBPedia OWL ontology. Assertions can be derived also from a database, or they can be included in the ontology directly. Some of the authors implemented various mapping strategies of assertions from databases (Pipitone et al., 2016a), (Pipitone and Pirrone, 2014) to be included in the model presented in this paper, but the description of such strategies is out of the scope of this work.

Formally, the domain ontology is the tuple $O = \langle C_o, P_o, T_s, L, P_d \rangle$ defined according to the W3C technical report specification[2], where:

- $C_o = \{cl_i\}$ is the set of type 1 classes;
- P_o is the set of the object properties, so that:

$$P_o = \{o_i \mid o_i = (cl_j, cl_k) \;\; cl_j, cl_k \in C_o\};$$

[2]https://www.w3.org/TR/owl-ref/

- $T_s = \{t_i\}$ is the set of literal datatypes;
- $L = \{l_i\}$ is the set of literal strings used in the ontology as values of t_i;
- P_d is the set of the datatype properties, so that:

$$P_d = \{d_i \mid d_i = (cl_j, l_k) \;\; cl_j \in C_o,\; l_k \in L\}.$$

The ontology formal definition provided here does not include individuals, that are the so-called *facts* or *instances*. In this model is considered the case where facts are obtained from a set of strategies for mapping assertions to the terminological structures, that are formalized by the *map* function so that $map : C_o \cup P_o \cup P_d \cup L \rightarrow I$, and $map(oe)$ returns the set I containing the instances for the ontological element oe defined by the assertion mapping.

3.4 Linguistic Knowledge

The linguistic base contains the set of constructions we defined for representing the linguistic typologies of a query in a specific local language. For example, in the applications described in Chapter 4, the model was tested to solve specific task using Italian and English language. In particular, considering that the objective of the proposed architecture is to answer to general questions about the domain, the typologies we referred to are the *direct real* interrogative sentences, which are related to something that is really unknown, and not to the *direct rhetoric* ones. Such a set is grouped in a taxonomy of constructions for implementing the corresponding continuum. The more general construction is the *direct interrogative*, which includes in order a particle, a verb, and what we called the *question topic*, that can be either a syntagma or a dependent clause.

The following code in a simplified FCG syntax (Steels, 2011b) shows the more abstract construction representing the direct interrogative, that is the top unit:

```
((?Top (sem-subunits (== ?particle ?verb ?questopic)))
(?particle (sem-cat (== (particle ?x))))
(?verb (sem-cat (== (verb ?x ?y ?z))))
(?questopic (sem-cat (== (questopic ?z))))
<->
((?Top (syn-subunits (== ?particle ?verb ?questopic)))
(syn-cat (==1 (pos (DI))))
(?form(form(==(meets ?particle ?verb)(meets ?verb ?questopic)))))
```

The double arrow separates the semantic pole and the syntactic one, which are written respectively over and under the arrow. The `meets` operator establishes an order between sub-units: in this case the particle comes before the verb, and the verb before the question topic. The question mark identifies variables that can be unified: `?particle`, `?verb` and `?questopic`. The `sem-cat` slot contains the semantic category for each variable. We defined slots purposely by trivial significance: `particle`, `verb` and `questopic`. Similarly, the `syn-cat` slot contains the syntactic category that is, in the case of the direct interrogative, the `DI` value. Some subsumed constructions might look as:

```
((?particle-unit  (meaning ((particle ?particle))))
<->
(?particle-unit (form ((syn-cat ADV))((string "Quando")))))
((label "annomorte" "annonascita" ...)))))
```

and

```
((?questopic-unit  (sem-subunits (== ?noun ?pre ?npr)))
(?pre (sem-cat (== (noun ?x))))
(?npr (sem-cat (== (npr ?y))))
(?pre (sem-cat (== (pre ?z))))
<->
((?questopic-unit (syn-subunits (== ?npr ?pre ?npr)))
(syn-cat (==1 (pos (SUBORDINATE))))
(?form (form (== (meets ?npr ?pre)(meets ?pre ?npr)))))
```

The first construction represents the particle that is the Italian adverb "*Quando*"; it is an item-based construction because the specific string form is indicated. In such a construction, the `label` slot is used to indicate the properties to be searched for in DBPedia to disambiguate the user request, as explained in section 3.5.1. These properties convey information about the question topic, and are linguistically related to the specified item. There can be different `label` slots for each item. For the sake of clarity, only one `label` is reported in the example.

The second example is another abstract construction, representing a question topic. Obviously, there are many kinds of question topic abstractions. In the example, the topic is the ordered conjunction of a proper noun, a preposition and another proper noun because this conjunction generally identifies the entity the user is questioning about. The particle and the verb that are unified next, define what piece of information is requested for the entity. For example, the phrase *Torre di Pisa*, which is POS tagged as (*Torre.NP di.PREP Pisa.NP*), is unified by means of the POS tags with both the generic constructions of the proper noun, and the item-based construction related to the preposition "*di*". Unification is then used to put constructions into conjunction to the more abstract question topic in the example. This strategy will be detailed in subsection 3.5.1. Summarily, we defined the sub-units of both general constructions and item-based ones where a string is associated, as in the case of adverbs, prepositions, conjunctions, and so on. These constructions are unified with the transient structure of the user question; if the user question does not contain any slot of the item-based constructions, more general constructions are unified. In this way, the model allows the comprehension of the query even if some of its part are unknown or incomplete.

3.5 Modeling the cognitive processes

In what follows a detailed description is provided regarding the two cognitive processes devised in QuASIt, that are the *conceptualization of meaning* and the *the conceptualization of form* along with their implementation in the proposed architecture.

3.5.1 Mapping to Meanings

The *Mapping to Meanings* module (MtM) implements the set of processes used by the agent to access the meanings of the natural language query formulated by the user. The query is first chunked by a POS tagger. A chunk is a set of consecutive tokens with the same POS tag; it is a n-gram of query words having the same syntactic category. Chunks are used for filling the transient structure that will unify with the constructions in the linguistic base; the semantic pole of this structure will be empty, while the syntactic one will contain two slots; the former will be related to the syntactic category of the chunk, the latter will contain its string form. The set of operations implementing the cognitive processes of agent's comprehension, are formalized in what follows.

Let consider the query $Q = \{q_1, q_2, ..., q_n\}$, where q_i is the i-th token. Being T the set of all POS tags, and $t \in T$ a specific tag, the chunks set C is the partition of Q so that:

$$C = \{c_i \mid c_i = \bigcup_{l}^{k} q_j,\ pos(q_j) = t_i\ \forall j \in [l \dots k]\,; l, k \in [1 \dots n]\,;\ t_i \in T;\ q_j \in Q\}$$

where the function $pos : Q \rightarrow T$ returns the POS tag of a token. The transient structure represents a sentence that has been understood partially; the MtM builds such a structure considering the chunks set of the query. The base process claims that *for each chunk a couple of slots is built in the transient structure, containing respectively the POS tag and the tokens of the chunk.* As an example, if the question is "*Quando è morto Riccardo Zandonai?*" the module outputs the following chunks expressed as set of bindings: {*(Quando . ADV)*, *(è. VP)*, *(morto . ADJ)*, *(Riccardo Zandonai . NPR)*}.

Formally, given the set C, the initial transient structure t_s is a couple $t_s = \langle sem, syn \rangle$ where $sem = \emptyset$, while syn is a set of couples $syn = \{$`(syn-cat` t_i`)(string` c_i`)`$,\ c_i \in C,\ t_i \in T\}$. The complete transient structure for the example is represented in figure 3.2.

The structure is then unified with the linguistic base. Through the unified constructions, a general query pattern emerges, and the semantic side is filled. In the example, the unified pattern is:

```
((?Top (sem-subunits (== ?particle ?verb ?questopic)))
(?particle (sem-cat (== (particle Quando))))
(?verb (sem-cat (== (verb morto))))
(?questopic (sem-cat (== (questopic Riccardo Zandonai))))
```

```
<->
((?Top (syn-subunits (== ?particle ?verb ?questopic)))
(syn-cat (==1 (pos (DI))))
(?form (form (== (meets ?particle ?verb)(meets ?verb ?questopic))
((syn-cat ADV) (string "Quando"))
((syn-cat V)(string "morto"))
((syn-cat NPR)(string "Riccardo Zandonai"))
))
```

where all query components emerge as result of a sequences of unification steps. Formally, we call F the store collecting all these components. In particular, $F = F_p \cup F_v \cup F_q$, where F_p contains the particle slots, F_v contains the verb slots, and F_q contains the question topic. In the example, $F_p = \{\texttt{Quando}\}$, $F_v = \{\texttt{morto}\}$ and $F_q = \{\texttt{Riccardo Zandonai}\}$. The question topic is what the user wants to know, and it is then searched for into the domain ontology to extract the corresponding *assertion subgraph*.

Extraction of the assertion subgraph corresponds exactly to conceptualization of the perceived words. In this work, extraction is simply implemented by rough intersections between the stems of the words in the unified query construction, and the labels of the domain ontology. Being *stem* the function that returns the stem of the words in its argument, *couple* the function that returns both elements of a couple, and f_i a generic element belonging to either F_q or F_v, the assertion extraction process is modeled by the following functions:

- $a_c : F_q \rightarrow C_o$ that returns the set:
  $a_c(f_i) = \{\ cl_k \mid stem(f_i) = stem(i_j),\ i_j = map(cl_k),\ cl_k \in C_o\}$
  composed by the ontological classes whose instances label's stems map to the stems of question topic words contained in F_q. The set $I = \{i_j\}$ will

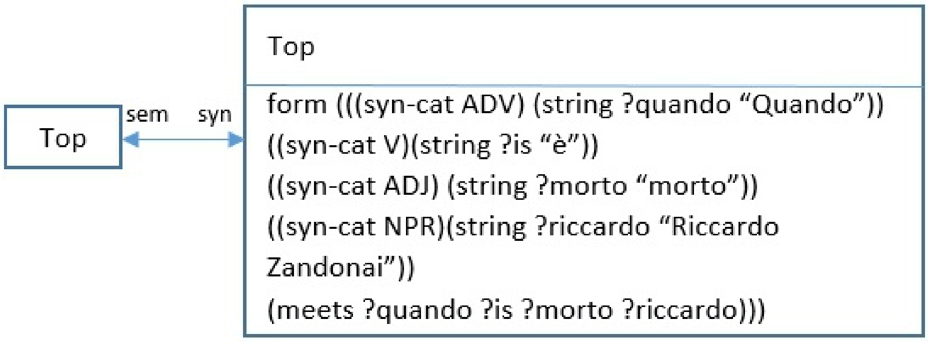

Figure 3.2: The initial transient structure corresponding to the query in the example.

contain all such instances;

- $a_p : F_v \rightarrow P_o \cup P_d$ returning the set:
  $a_p(f_i) = \{\ p_j \mid stem(f_i) = stem(p_j),\ p_j \in P_o \cup P_d \wedge couple(p_j) \supset a_c(f_i)\}$
  composed by the properties of the classes in $a_c(f_i)$ whose label's stems map to the stems of the verb in F_v.

The result is a sub-ontology $A = \langle C_a, P_a, I \rangle$ where $C_a = \bigcup_{F_q} a_c(f_i)$, $P_a = \bigcup_{F_v} a_p(f_i)$, and I the set of individuals retrieved by a_c. Its worth noting that in this case the ontology definition includes the instances. In the proposed example, $C_a = \{$`Person`$\}$, $I = \{$`Riccardo Zandonai`$\}$ (the map returned by a_c is exactly this instance). Properties result to be $P_a = \{$`luogomorte`, `annomorte`$\}$ that are connected to nodes with the same names in DBPedia; these are the properties whose stems match to the stem of the verb specified in the unified query pattern ($F_v = \{$`morto`$\}$). To complete the comprehension process, QuASIt prunes further the properties in the subgraph by the `label` slot annotated in F_p. In the example, QuASIt prunes the `luogomorte` property because it is not annotated in the `label` slot of the adverb *Quando*.

3.5.2 Mapping to Forms

Once the assertion subgraph has been retrieved QuASIt runs the processes aimed at simulating lexical access in human mind. Here the correct expressions related to the comprehended concepts must be formulated. The structure of the answer depends on the type of question; the proposed model can accept both essay questions and multiple choice ones. In the first case the correct answer has a free structure, while in the second case it must be one of the proposed candidates. Even if the two cases will be described separately, both of them share the same strategy to retrieve information from the support text, if available.

Searching in the support text

Searching in a support text is a possible strategy to deal with unstructured information when an artificial agent is trying to answer a particular question, it represents the way according to which the agent attempts to retrieve an information that is not in its knowledge base, trying to learn a possible answer by comprehending a plain text dealing with the question topic. Such process is implemented in QuASIt by extracting from the text the sentence with the highest syntactic similarity to the one owned by the agent; such a sentence can be either the query itself or one of the multiple answers. In fact, the query and the multiple choices represent the only source of information the agent owns to devise the correct answer when it has no coded knowledge about the question topic.

Formally, let $Q = \{q_1, q_2...q_n\}$ be the query of the user, and $P = \{p_1, p_2, ...p_m\}$ a sentence in the support text; each element in these sets is a token. P will be considered as much similar as Q when maximizing the following similarity measure m:

$$m = |\Im| - (\alpha l + \beta u) \tag{3.1}$$

where $\Im = \{p_j |\ \exists q_i \in Q,\ J(p_j, q_i) > \tau\}$, and $J(p_j, q_i)$ is the Jaro-Winkler distance between a couple of tokens (Winkler, 1990). As a consequence, $\Im \supset Q \cap P$, and $|\Im|$ is the number of matching tokens both in Q and P.
$l = 1 - \frac{|\Im|}{|P|}$ is the number of "lacking tokens" that are tokens belonging to Q that do not match in P, while $u = 1 - \frac{o(Q,\Im)}{|\Im|}$ is the number of "unordered tokens" that is the number of tokens in Q that do not have the same order in $\Im$; here $o(a, b)$ is the function returning maximum number of ordered tokens in a with respect to b.

Both l and u are normalized in the range $[0 \ldots 1]$; they are penalty values representing syntactical differences among the sentences. The higher u and l are, the lower is the sentences similarity.

The α and β parameters weight the penalty, and they have been evaluated empirically through experimentation along with τ.

Essay questions

In case of essay questions, QuASIt produces the assertions that are contained in the range of the properties belonging to the assertion subgraph, which are separated by comma and conjunctions. Right now, we do not produce a formal answer as the affirmative form of the direct interrogative. Assertions are those connected to the properties in the a_p set by the MtM strategies. If a_p is empty, QuASIt does not know the question topic, and it starts searching for external information sources. The support text contained in the DBPedia `Abstract` property related to the question topic, and the one contained in the question itself, if available, are searched for to extract the sentence $\widehat{P}$ that maximizes the similarity with respect to the question string. In this case the plain n value is used because the question string has an obvious different order with respect to the sentences in the support text, and it contains much less tokens.

Multiple choice questions

QuASIt can be used for choosing the correct answer to a question in a set of candidates. These candidates represent the sentences owned by the system, and they are managed according to the strategies explained above. In particular, two steps are executed: first QuASIt searches for each candidate in the assertion subgraph, filtering the best matching answer according to the metric m.

If no candidates match to any assertion, QuASIt refers to the available support text, that is the concatenation of the DBPedia `Abstract` property related to the question topic and the support text enclosed in the question itself, if available. Again QuASIt searches for the best matching candidate to some sentence in the support text according to the similarity value m. The form of the answer is directly the winning candidate.

Chapter 4

Applications

In this section all the applications of the QuASIt agent are described for specific NLP tasks. The first tool is ChiLab4It (Pipitone et al., 2016b), a customization of QuASIt to address the task of selecting the most relevant FAQ among those contained in a given *FAQ base* according to the question asked by the user. The second one (Pipitone et al., 2017) is a system aimed at performing the NEEL task on informal text sources like tweets.

4.1 ChiLab4It

ChiLab4It uses the functions of QuASIt aimed at answering multiple choice questions using a support text, to understand and choose the most relevant FAQ in a *FAQ base*, given a query from the user. A FAQ can be regarded exactly as a support text, that can be used to understand the query sentence and to provide the answer. Moreover this tool enhances the sentence similarity measure introduced in the QuASIt cognitive model in two ways. First, three separate measures are computed for the three parts of a FAQ that is question text, answer text and tag set, and they are summed to provide the final similarity. Second, the synonyms of the query words are analyzed to match the query against each sentence of the answer text of the FAQ to achieve linguistic flexibility when searching for the query topic inside each text.

ChiLab4It was tested with the QA4FAQ@EVALITA2016 (Caputo et al., 2016) competition data, and it resulted to be the winner having a c@1 rank well above the fixed experimental baseline and reaching the best score.

4.1.1 Implemented strategy

The strategy implemented in ChiLab4It system is based on the QuASIt function that selects the correct answer to multiple choice questions using support text; the intuition was that *a FAQ can be considered a support text* that can be used for retrieving the more relevant FAQ to a user's query. In ChiLab4It the α and β parameters shown above are set with different values depending on which kind of support text is considered during the search, as next explained. In this section will be shown this strategy in detail, and next how it is applied in the proposed tool.

The basic idea was to consider a FAQ as a support text. According to the provided dataset, a FAQ is composed by three textual fields: the *question text*, the *answer text* and the *tag set*. For each of these fields we applied the search strategy defined above; in particular we set different α and β parameters for each field in the m measure reported in equation 3.1, depending on linguistics considerations. For this reason, we defined three different parameterized m measures named m_1, m_2 and m_3. Moreover, further improvements were achieved by searching for the synonyms of the words of the query in the answer text. These synonyms were not considered in the QuASIt implementation.

Given the previously defined variables $\Im$, l and u, the α and β parameters were set according to the following considerations:

- *question text*; the α and β parameters are the same of QuASIt, that is $\alpha = 0.1$ and $\beta = 0.2$. This choice is based solely on linguistic motivations; in fact, considering that the support text is a question such as the user query, both sentences to be matched will have interrogative form. As a consequence, both l and u influence the final match. The final measure is:

$$m_1 = |\Im| - (0.1 * l + 0.2 * u)$$

- *answer text*; the search is iterated for each sentence in the text. In this case, the α and β parameters are zero ($\alpha = 0$ and $\beta = 0$). This is because the answer text has a direct form, so the order of tokens must not be considered; moreover, a sentence in the answer text owns more tokens than the query, so this information is not discriminative for the final match.
  In this case, *the search is extended to the synonyms of the words in the query* except to the synonyms of the stop-words; this extension has improved significantly the performances of the system. Empirical evaluations demonstrated that there were not the same improvements when the synonyms were considered for the other parts of a FAQ (question text and tag set) because in these cases the synonyms increase uselessly the number of irrelevant FAQs retrieved by the system.
  Formally, let Σ be the σ-*expansion* set (Pipitone et al., 2014) that contains both the words and the synonyms of such words in the $Q - S_w$ set, being Q the user query as previously defined and S_w the set of stop-words:

$$\Sigma = \{\sigma_i \mid \sigma_i = synset(q_i) \ \wedge \ q_i \in Q - S_w\}$$

Let's define $S = \{S_1, S_2, \ldots, S_N\}$ the set of sentences in the answer text. We defined the M set that contains the m_{s_i} measures computed with $\alpha = 0$ and $\beta = 0$ in m, for each sentence $S_i \in S$ with the σ-expanded query:

$$M = \{m_{s_i} \mid m_{s_i} = |\Im_i|\}$$

where

$$\Im_i = \{p_j \in S_i \cap \Sigma \mid \exists q_k \in Q,\ J(p_j, q_k) > \tau\}$$

The final similarity measure m_2 will be the maximum value in M:

$$m_2 = max\ \{m_{s_i} \mid m_{s_i} = |\Im_i|\}$$

- *tag set*; the α and β parameters are zero ($\alpha = 0$ and $\beta = 0$) also in this case. This is because the tags in the set do not own a particular linguistic typology, so the information related to both the order of tokens and the lacking ones must not to be considered. As already explained, the synonyms are not included in this search. As consequence:

$$m_3 = |\Im|$$

  where $\Im$ is the previously defined intersection among the query of the user and the set of tags.

A query will be considered as much similar as a FAQ when maximizing the sum of the measures defined previously, so the final similarity value is:

$$m_{faq} = m_1 + m_2 + m_3$$

These values were ordered, and the best FAQ is outputted for a single query.

4.1.2 The architecture

In figure 4.1 the architecture of ChiLab4It is shown; the input is the query of the user, while the output is a list of the most relevant FAQs. The sources became the *FAQ base* and the *Wiktionary* source from which the provided FAQ dataset and the synonyms are respectively queried.

The white module of such an architecture is the MtF module as implemented in QuASIt. The dark modules are the integrations that have been applied to the MtF module for customizing it to the FAQ domain; in particular, such integrations regard both the σ-expansion of the query and the setting of the analytic form (including parameters) of the m measure depending on the FAQ field.

The first integration is implemented by the σ *module*, that returns the Σ set for the query of the user retrieving the synset from Wiktionary[1].

Parameters and the measure settings are performed by the *FAQ Ctrl* module which is encapsulated into the main MtF module; it retrieves the FAQ from the *FAQ base* and customizes the m measure according to the analyzed field (m_1 for the question text, m_2 for the answer text, m_3 for the tag set). The MtF module computes such measures referring to the σ-expanded query, and finally the m_{faq} value is computed and memorized by the *FAQ Ctrl* for tracing the id of the FAQ with the highest value.

[1] https://it.wiktionary.org/

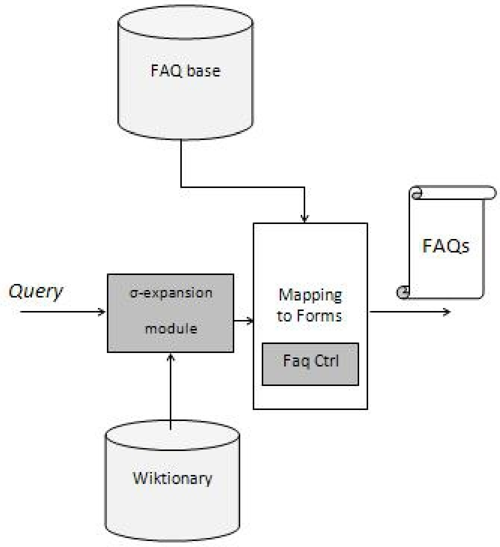

Figure 4.1: The ChiLab4It Architecture

4.1.3 A practical example

In this section is shown a toy example with the aim of explaining better the searching process in the support text and how the similarity measure works.

Such an example is a real question as retrieved in the data set provided by the organizers of QA4FAQ@EVALITA2016 (Caputo et al., 2016), that is the competition used to test the effectiveness of this application.

Let consider the query with $id = 4$, that is: "*a quali orari posso chiamare il numero verde*". In this case, the Q and the S_w set are:

$$Q = \{A,\ quali,\ orari,\ posso,\ chiamare,\ il,\ numero,\ verde\}$$

and

$$S_w = \{A,\ il\}$$

being "*a*" and "*il*" the stop-words in the question.

⟨**faq**⟩
⟨**id**⟩339⟨/**id**⟩

⟨**question**⟩Quali sono gli orari del numero verde?⟨/**question**⟩

⟨**answer**⟩Il servizio del numero verde assistenza clienti AQP 800.085.853 e attivo dal lunedi al venerdi dalle ore 08.30 alle 17.30, il sabato dalle 08.30 alle 13.00; il servizio del numero verde segnalazioni guasto 800.735.735 e attivo 24 ore su 24.⟨/**answer**⟩

⟨**tag**⟩informazioni, orari, numero verde⟨/**tag**⟩
⟨/**faq**⟩

Table 4.1: The XML description of a FAQ as provided in the data set

The highest measure is computed by ChiLab4It in correspondence to the FAQ shown in table 4.1. Considering this FAQ, let compute the three measures for the *question text*, the *answer text* and the *tag set*.

In the first case the support text is the question text of the FAQ, and the P set is:

$$P = \{Quali,\ sono,\ gli,\ orari,\ del,\ numero, verde\}$$

with $|P| = 7$. The m_1 value will be computed considering that the intersection $\Im$ between the question text and the query of the user is:

$$\Im = \{quali,\ orari,\ numero,\ verde\}$$

The Jaro-Winkler distance is 1 for each word, and $|\Im| = 4$. Also, $l = 1 - \frac{|\Im|}{|P|} = 1 - \frac{4}{7} = 0.428$. For the calculation of u, we notice that $o(Q, \Im)$ returns 4 because the tokens in Q are all ordered with respect to $\Im$, that means they follow the same sequence in $\Im$. As consequence, $u = 1 - \frac{o(Q,\Im)}{|\Im|} = 1 - \frac{4}{4} = 0$. Substituting all values, m_1 will be:

$$m_1 = |\Im| - (0.1 * l + 0.2 * u) = 3.95$$

In the next step, we consider the answer text; in the FAQ, this text is composed by only one sentence that becomes the new support text P, and the procedure will be applied once. In particular, $S = \{S_1\}$ and $P = S_1 = \{$*Il, servizio, del, numero, verde, assistenza, clienti,...., attivo, 24, ore, su, 24*$\}$ as shown in table 4.1. In this case, the m_2 measure depends only from the intersection between the σ-expanded query and S_1. In particular, the Σ set is computed unifying the difference set $Q - S_w = \{$*Quali, orari, posso, chiamare, numero, verde*$\}$ with the synset from Wiktionary of each such token, so: $\Sigma = \{$*[[quali], [orari], [posso], [chiamare, soprannominare, chiedere, richiedere], [numero, cifra, contrassegno numerico, matricola, buffone, pagliaccio, elenco, gruppo, serie, classe, gamma, schiera, novero, taglia, misura, attrazione, scenetta, sketch, esibizione, gag, sagoma, macchietta, fascicolo, puntata, dispensa, copia, tagliando, contrassegno, talloncino, titoli, dote, requisito], [verde, pallido, smorto, esangue, acerbo, giovanile, vivace, vigoroso, florido, verdeggiante, lussureggiante, rigoglioso, agricolo, agrario, vegetazione, vigore, rigoglio, freschezza, floridezza, via, avanti, ecologista, ambientalista, livido]]*$\}$, where the synsets are represented in square brackets for clarity. The intersection $\Im_1 = \Sigma \cap S_1$ is simple $\Im_1 = \{$*numero, verde, orari*$\}$ because these tokens have the highest Jaro-Winkler distance from the tokens in S_1. As consequence, $M = \{|\Im_1|\} = \{3\}$ and $m_2 = 3$.

In the third case, the support text is the tag set, so $P = \{$*informazioni, orari, numero, verde*$\}$ and $\Im = \{$*orari, numero, verde*$\}$. The m_3 value is simply $m_3 = |\Im| = 3$.

Finally, the m measure is computed adding the three calculated values, so

$m = 3.95+3+3 = 9.95$ that represents the highest value among those computed for all FAQs in the dataset.

4.2 Using the QuASIt model for NEEL

The proposed application attempted to perform the semantic annotation of tweets applying the model previously explained. The tool is inspired to the cognitive model already proposed for QA (Pipitone et al., 2016c) to overcome the problems posed by informal language, and apply the system to NEEL in tweets.

As seen, such an approach does not perform statistical inference and it is not based on any machine learning paradigm, rather it attempts to simulate the cognitive processes that take place in humans when they infer the correct answer. Processes are encoded through the rules and the similarity measure used to link the query to the ontology: as a result, the 'semantic sense' of the query is inferred, and the answer is produced.

Under this perspective, it is defined the 'web sense' of the tweet by creating hyperlinks between the tweet itself and the linked data source DBPedia. The similarity measure and the cognitive processing rules are properly re-defined for highlighting the particular kind of both the text, that is a tweet with its typical features, and the task. In a few words, the main research contribution of this application is the definition of new aspects in the implementation of the cognitive processes to allow linking a tweet's informal text to DBPedia; such processes consider the particular structure of a tweet (mentions, hashtags, and partially structured statements) and the nature of the NEEL task. The whole approach is based also on linguistic considerations about the informal language.

4.2.1 Background considerations

When the social data are tweets, the main problems arise in both the informal language and shortness of the text (Rizzo et al., 2016): the use of a ‘loose’ language with abbreviations, sparsity of contents and not enough precise context in place of a formal one, are serious obstacles for the typical semantic annotation approaches. Moreover, the shorter is the text, the worse are results produced by the context disambiguation, and the annotation can fail. A grammatical evaluation of the informal language may be useful for solving many of these issues but such a grammar must not be a formal and strict one. Hand-crafted grammar-based systems typically produce good results, but at the cost of hard work for manual annotations by experts in Computational Linguistics (CL). Moreover, if the systems perform statistical evaluations, they typically require a large amount of training data too. Semisupervised approaches have been suggested to avoid part of the annotation effort as at (Nothman et al., 2013), but they do not exhibit better performances than the previous ones.

4.2.2 NEEL in tweets

A tweet is a post on the social media application Twitter for which we identified four main components in the structure, that are:

1. the *micropost*, that is the message shared by the user. It can not be longer than 140 characters;
2. the *hashtag*, that is each metadata tag with the # prefix, that allows to categorize a tweet’s topic(s), and then makes it easier for users to search other tweets about such topic(s);

3. the *tag*, that is the metadata tag with the @ prefix, that allows to associate an entity (other people, locations and so on) already existing in the social platform;

4. the *thread*, that are the set of comments of other people to the main micropost.

According to (Rizzo et al., 2016) semantic annotation of tweets consists of two main operations: *mention detection* and *candidate selection*. The former is related to the identification of the entity mention in the tweet, the latter is the operation related to the identification of the link in DBPedia that defines such an entity. We devised some adjustments to the cognitive processes of QuASIt with the aim to model these operations. The proper nature of the NEEL task poses some constraints in the elaboration of a tweet, and if the task is executed by a human, she/he should keep in mind these constraints thus acting inside these boundaries. The NEEL constraints that we used to model our system are:

- a mention of an entity in a tweet is a proper noun or an acronym;
- the complete extent of an entity is the entire string representing the name, without any pre-posed (i.e. the articles, the title such as "Mrs", "Dr", and so on...) or post-posed modifiers. The sub-strings of an extent, if exist, are not considerable as single entity mention; for example, the mention "Micheal Jackson" is a complete extent of a single entity, while the sub-words "Michael" and "Jackson" are not single mentions. These words can be considered as single mentions when they are the only string in the extent;
- an embedded entity must be considered an entity mention, while the broader one not. An embedded entity is encapsulated into a more generic

one (the broader one) that is not explicitly mentioned; for example, in the statement "The art director of Harry Potter", the extent "Harry Potter" is an embedded entity that must be annotated, while "The art director person" is the broader entity, and must not be annotated;

- the words in the hashtag ('#') or in the tag ('@') are entities only if they are proper nouns or acronyms.

4.2.3 Cognitive processes for NEEL in tweets

The key of the proposed strategy was to consider a tweet as a query to be understood by the QuASIt system. As consequence, the *conceptualization of meaning* process has been applied for understanding a tweet, hence to link the tweet to the ontology. New linguistic aspects have been considered due to the informal language and the particular structure of tweets, and considering the NEEL task nature. The processes defined for the NEEL task are:

- chunking activity;
- chunk conceptualization;

These processes implement the NEEL operations previously described; in particular, the chunk conceptualization process is an insight of the conceptualization of meaning in QuASIt for allowing NEEL. As a consequence, the QuASIt architecture was modified as shown in figure 4.2.

Chunking activity

To perform mention detection in a tweet, chunking is the base cognitive process performed by humans for understanding a not plain text (Rupley et al.,

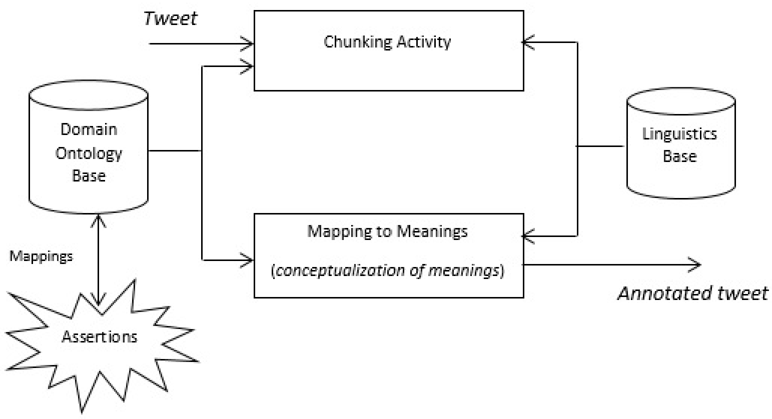

Figure 4.2: The Cognitive Architecture for NEEL in tweets

2009). Humans break down a difficult text into more manageable pieces, and rewrite such "chunks" in their own words. Given the tweet t, we classified its components into two main categories in terms of the linguistic properties of the inherent chunks. The categories are:

1. the M category, where $M = \{m_i \mid m_i \text{ is a micropost of } t\}$; such category contains the microposts of the tweet, comprising the main micropost that generates a discussion, and the microposts in the thread that are the comments to the main one;

2. the H category, where $H = \{h_i \mid h_i \text{ is a hashtag or a tag of } t\}$, that contains the hashtags and the tags in t.

Trivially, both hashtags and tags in M require a more deep chunking activity than microposts, for which blank spaces and/or punctuation separate tokens.

Chunking for tags and hashtags is implemented using a cognitive inspired approach: here the linguistic problem can be stated as obtaining a "meaningful" splitting of a word concatenation where there are no separation characters. The term "meaningful" has to be intended as "finding the words, which make more sense while taking into account morphology", and it will be the key for selecting between multiple chunks. The process has been implemented by means of the A^* semantic tokenizer proposed by some of the authors at (Pipitone et al., 2013). A tree of possible alternatives for chunking a concatenated string is built by means of proper heuristics relying on linguistic considerations that set proper cost functions too. A left-to-right scanning is used to model the reading activity, that was inspired to the Simple View of Reading (SVR) (Hoover and Gough, 1990). Input strings are scanned from left to right and candidate words are generated by subsequent character concatenations from the very first character to the entire string. The tree is explored by best-first search, and the path with lower cost links nodes that correspond to the identified tokens.

In this approach, differently from the original tokenizer, the source we used for retrieving meaningful words is formed by both DBPedia ontology and Wordnet linguistic source (Fellbaum, 1998c). In fact, the tags and the hashtags can contain proper nouns that are not in Wordnet even if they are referred to in DBPedia, and vice versa. If no sense words are retrieved, the whole string is returned. After chunking the text in H, the microposts in M are chunked too, performing at first a simple tokenization based on blank spaces; due to the presence of informal language, such tokens are chunked again according to the A^* strategy. For completing the chunking activity process, the chunks devised so far must be rewritten using words already owned by the system. In particular, considering the informal language, such an operation implies the substitution of ill-formed syntax words with well-formed ones that are in the sources used by the system, and represent its language knowledge. Such a substitution is performed considering the words that are syntactically nearest to the token; more simply, an automatic corrector[2] based on Wordnet is applied to the tokens, and a list of similar words is returned. Formally, let be $a\_star(s)$ the function that returns the list of chunks L_c for the string s based on the $A*$ strategy; $tok(s)$ is the function that returns the list of tokens L_t split using blank spaces for the string s; $icbl(k)$ is the function that returns the list L_w of syntactically similar words to the token k. The chunking activity process is modeled by the following functions:

$$c_a : H \cup M \to L_c, \;\; c_a(s) = \begin{cases} a\_star(s) & s \in H \\ a\_star(tok(s)) & s \in M \end{cases}$$

$$icbl : L_c \to L_w, \qquad icbl(k) = \{w_i \mid w_i \text{ is a word syntactically similar to } k\}$$

The output of the chunking activity is the set $C = H \cup M \cup L_w$ composed by all the possible words to be analyzed for mention detection. Next, a POS tagger allows filtering such words to identify only the proper nouns, and hence the *topic* of the tweet; differently from QuASIt, where the topics were the noun phrases contained in the query (both common and proper nouns), in this case the topics are only the proper nouns in the chunk set C according to the NEEL task specifications defined previously.

Finally, being w a generic set of words, $pos(w)$ the function that returns the set of POS tags of the words in w according to the chosen tagset[3], the mention detection process ends with the definition of the set MD that will contain all candidate mentions:

$$MD = \{m_i \mid m_i = \{c_j, c_{j+1}, ..., c_{j+n_i}\} \subset C, \; pos(m_i) \in \{NP, NPS\}\}.$$

[2]Downloadable at `https://www.icbld.com/`

[3]Specified at `https://courses.washington.edu/hypertxt/csar-v02/penntable.html`

The n_i value represents the extent of the i-th mention, and NP and NPS are the tags associated to the proper noun (respectively singular and plural).

Chunk conceptualization

Once the set MD containing all the possible candidate mentions is built, the chunk conceptualization process attempts to select from DBPedia the entities to link to such mentions. Such a process is based on the conceptualization of meaning defined in QuASIt, particularly in the part related to the extraction of the assertion subgraph.

The assertion subgraph corresponds exactly to the conceptualization of a perceived statement because it links the statement itself to the ontology used by the system; its extraction is now implemented for the tweet. The rough intersections between the stems of the mentions in MD, and the stems of the labels in DBPedia is extended here considering a suitable syntactic similarity and not a perfect match. Recalling the definitions for the functions *stem* and *map*, and the formal definition of the ontology given in section 3.3, we define the new function $sim(w_1, w_2) = 0.5 * jaro(w_1, w_2) + 0.5 * lev(w_1, w_2)$ that returns a similarity measure that is a weighted sum of the Jaro-Winkler $jaro(\cdot\ ,\ \cdot)$ and Levenstein $lev(\cdot\ ,\ \cdot)$ distances between the words w_1 and w_2 in its argument. The use of a string similarity metric enhances the mapping process taking into account also ill-formed words in informal language that could make the *stem* function returning no results, thus resulting in an empty intersection with the stems coming from the labels of DBPedia. In this case, the Jaro-Winkler distance is computed from the original forms of the chunks; the chunks of a mention have to be concatenated by the *concat* function. The chunk conceptualization process is formalized by the function $a_{c_{neel}} : MD \rightarrow C_o$ that returns the set:

$$a_{c_{neel}}(m_i) = \{\ cl_k \mid stem(m_i) = stem(i_j) \vee$$
$$sim(concat(m_i), i_j) > \tau,\ i_j = map(cl_k),\ cl_k \in C_o\}$$

composed by the ontological classes such that either the stems of their instance labels is equal to the stems of mentions in MD or the Jaro-Winkler distance between such labels and the form of mentions in MD is above a suitable threshold. The value for τ has been fixed to 0.7 experimentally. The set $I = \{i_j\}$ will contain all such instances.

Candidate selection

The candidate selection process is modeled by the $a_{c_{neel}}$ function that returns for each mention in MD the entities in DBPedia which better match to it, according to the criterion specified in the function definition. Finally, the set of nodes C_a and the set of correspondent instances I, where $C_a = \bigcup_{MD} a_{c_{neel}}(m_i)$, and I the set of individuals correspondent to $a_{c_{neel}}$ represent what the system understands about the whole tweet, that is its assertion subgraph.

Chapter 5

Main Experiments

In this chapter the main experiments will be reported aimed at testing the effectiveness and efficiency of the QuASIt agent and its customizations when engaged in the tasks described in the previous chapters.

At first the results achieved by the *pure* QuASIt system will be reported, using both Italian and English data sets made by questions with multiple option answer. employed in the past QA4MRE competitions held at CLEF conferences.

Next, the results achieved by ChiLab4It when participating at the QA4FAQ competition inside the EVALITA2016 conference. In this task ChiLab4It was engaged in Italian FAQ retrieving to answer the user requests. The system, gained the best score among the other participants, and it was the only one to outperform the proposed baseline.

Finally, the results achieved by QuASIt will be reported, and compared with the other state of the art tools when employed in NEEL task on informal English reported in tweets. Such data derive from the #Micropost2016 workshop NEEL Challenge co-located with the World Wide Web conference 2016 (WWW '16).

5.1 Testing QuASIt using multiple-choice questions

5.1.1 Dataset and task description

As stated above, QuASIt is able to accomplish question answering both providing an open answer and using multiple choices given by the user.

Due to the poor number of tests and tools doing these tasks for Italian, it was hard to find a way of benchmarking our system.

The most suitable data sets turned out to be those provided by the QA4MRE 2011 and QA4MRE 2012 competitions, and are made of multiple-choice questions. In this way the system can be tested in a comparable way with other state of the art systems. The task focuses on reading a single document, and identifying the answers to a set of questions about the information that is stated or implied in the text.

Going into more detail, these data sets consist of 120 and 160 questions respectively, each with 5 possible answers where just one is correct. Questions are grouped by topic and, for each topic, support texts are provided containing the information about the related group of questions.

Each correct answer is specifically designed to require various kinds of inference and the additional knowledge obtained through a background document collection may be used to assist with answering the questions in union with the provided support text.

Questions may be of the following types:

- FACTOID: Where or when or by whom

- CAUSAL: What was the cause/result of Event X?
- METHOD: How did X do Y? Or: In what way did X come about?
- PURPOSE: Why was X brought about? Or: What was the reason for doing X?
- WHICH IS TRUE: Here one must select the correct alternative from a number of statements, e.g. What can a 14 year old girl do?

5.1.2 Metrics and evaluation

The main measure used in this evaluation campaign is c@1, which is defined in equation 5.1.

$$c@1 = \frac{1}{n}(n_R + n_U \frac{n_R}{n}) \tag{5.1}$$

where:

n_R : is the number of correctly answered questions,

n_U : is the number of unanswered questions,

n : total number of questions.

As explained in the system description, the application performs a free search in the support text using a weighted scoring method. Consequently, tests are performed varying the score weights from 0 to 1 in order to optimize the system accuracy, and considering the performance variation of the system in term of its accuracy, defined by:

$$accuracy = \frac{n_R}{n}$$

Table 5.1: Accuracy varying weights

	$\alpha = 0.0$	0.1	0.2	0.3	**0.4**	0.5	0.6	0.7	0.8	0.9
$\beta = 0.0$	0.23	0.23	0.23	0.23	**0.22**	0.23	0.23	0.23	0.23	0.23
0.1	0.24	0.24	0.24	0.24	**0.23**	0.24	0.24	0.24	0.24	0.24
0.2	0.24	0.24	0.24	0.24	**0.23**	0.24	0.25	0.24	0.24	0.24
0.3	0.24	0.25	0.24	0.24	**0.24**	0.24	0.25	0.24	0.23	0.24
0.4	0.24	0.25	0.24	0.24	**0.24**	0.24	0.24	0.23	0.23	0.24
0.5	0.24	0.25	0.24	0.24	**0.24**	0.24	0.24	0.23	0.23	0.24
0.6	0.24	0.25	0.25	0.24	**0.26**	0.26	0.24	0.23	0.23	0.23
0.7	**0.24**	**0.25**	**0.25**	**0.24**	**0.26**	**0.25**	**0.24**	**0.23**	**0.23**	**0.23**
0.8	0.24	0.25	0.25	0.26	**0.26**	0.26	0.24	0.24	0.23	0.23
0.9	0.24	0.24	0.24	0.24	**0.25**	0.25	0.25	0.23	0.23	0.22

Table 5.2: Test Results

Test Dataset	Correct	NoA	Total	Accuracy	c@1
QA4MRE2011	40	2	120	0.33	0.33
QA4MRE2012	46	1	160	0.29	0.29

These experiments are conducted considering the support text as the only source of knowledge for the system, since weights are not implied in scoring answers coming from other used knowledge bases except those found in the support text.

From the experimental results reported in table 5.1 testing the system in both data sets, the best results are achieved using $\alpha = 0.4$ and $\beta = 0.7$ values.

Considering these as the optimal weights, the system was tested in both data sets using knowledge provided by both ontological resources and support text.

Looking at the table 5.2, the system achieves a c@1 score of 0.33 in the first dataset and 0.29 in the second. The equality between accuracy and c@1 values is explained because the system is not able to give an answer only for a negligible number of questions.

In comparison with other systems, referring to the past campaigns we had no participants for Italian in 2011, while only one participant there was in 2012.

In particular, as stated in (Peñas et al., 2012) such a team tested their system on both data sets achieving an average $c@1$ value 0.30. Furthermore, this system reached $c@1 = 0.35$ for the 2012 campaign with particular dataset dependent tuning.

5.2 ChiLab4It Testing and Evaluation

5.2.1 Dataset and evaluation task description

The dataset used for the evaluation of ChiLab4It system was the one provided by the QA4FAQ task organizers (Caputo et al., 2016). The system itself was submitted to the QA4FAQ@EVALITA2016 challenge in order to compare it with other state of the art systems. At the end of competition, ChiLab4It was classified as the best system among all the participants and the only one able to go beyond the baseline set by the organizers. Searching within the Frequently Asked Questions (FAQ) page of a web site is a critical task: customers might feel overloaded by many irrelevant questions and become frustrated due to the difficulty in finding the FAQ suitable for their problems. Perhaps they are right there, but just worded in a different way than they know. The QA4FAQ task consists in retrieving a list of relevant FAQs and corresponding answers related to the query issued by the user (Caputo et al., 2016).

The organizers released such a dataset as a collection of both questions and feedbacks that real customers provided to the "AQP Risponde" engine [1] set up

[1] `http://aqprisponde.aqp.it/ask.php`

by the Acquedotto Pugliese Company.

In particular, such dataset includes:

- a knowledge base of about 406 FAQs, each composed by the text fields we referred to;
- a set of query by customers;
- a set of pairs that allows organizers to evaluate the possible contestants. The organizers analyzed the feedbacks provided by real customers of the AQP Risponde engine, and checked them for removing noise.

The participants must provide results in a text file. For each query in the test data, the participants can provide 25 answers at the most, ranked according by their systems. Each line in the file must contain three values separated by the TAB character: *<queryid><faqid><score>*.

Training data were not provided: indeed, AQP was interested in the development of unsupervised systems, like ChiLab4It is.

5.2.2 ChiLab4It results and discussion

According to the guideline, we provided results in a text file purposely formatted, and for each query in the dataset we considered the first 25 answers. However, only the first FAQ is considered relevant for the scope of the task.

ChiLab4It is ranked according to the *accuracy@1* (c@1), whose formulation is reported in equation 5.1.

A participant could have provided two different runs, but in our case we considered only the best configuration of the system. In table 5.3 we show the

Table 5.3: The final results for QA4FAQ@EVALITA2016 task

System	c@1
ChiLab4It	**0.4439**
baseline	*0.4076*
fbk4faq.2	0.3746
fbk4faq.1	0.3587
NLP-NITMZ.1	0.2125
NLP-NITMZ.2	0.0168

final results with the ranks of all participants as provided by the organizers; our tool performed better than the other participants, and it was the only one that outperforms the experimental baseline.

The good performance obtained by ChiLab4It proves the effectiveness of question answering in the FAQ domain.

In Table 5.4 are reported some information retrieval metrics for each of the participating systems. In particular, Mean Average Precision (MAP), Geometrical - Mean Average Precision (GMAP), Mean Reciprocal Rank (MRR), Recall after five (R@5) and ten (R@10) retrieved documents, are computed. Moreover, the success_1 value is reported that is equal to c@1 without taking into account answered queries. It is possible to notice that on retrieval metrics the baseline is the best approach. This was quite expected since an information retrieval model tries to optimize retrieval performance. Conversely, the best system according to success_1 is ChiLab4It, that is based explicitly on a question answering approach, since it tries to retrieve a correct answer in the first position. This result suggests that the most suitable strategy in this context is to adopt a question answering model, rather than to adapt an information retrieval system.

Table 5.4: Other information retrieval specific metrics

System	**MAP**	**GMAP**	**MRR**	**R@5**	**R@10**	**success1**
ChiLab4It	**0.5149**	**0.0630**	**0.5424**	**0.6485**	**0.7343**	**0.4319**
baseline	0.5190	0.1905	0.5422	0.6805	0.7898	0.4067
fbk4faq.2	0.4666	0.0964	0.4982	0.5917	0.7244	0.3750
fbk4faq.1	0.4473	0.0755	0.4781	0.5703	0.6994	0.3578
NLP-NITMZ.1	0.3936	0.0288	0.4203	0.5060	0.5879	0.3161
NLP-NITMZ.2	0.0782	0.0202	0.0799	0.0662	0.1224	0.0168

5.3 Testing QuASIt in NEEL task

In this section we describe the challenge dataset and the evaluation scores that we referred to for evaluating the QuASIt model in NEEL task using informal language messages. Then the results will be shown and discussed in comparison to the other state of the art systems.

5.3.1 The dataset and the metrics

The dataset proposed for the #Microposts2016 NEEL Challenge extends the ones presented in the previous editions of the challenge with the only change in the DBPedia version that is DBPedia 2015-04. Particular attention was devoted to include both event and non-event tweets. Both Training and Development sets were not used to perform our experiments, because our system uses a symbolic unsupervised approach that can be applied in every context and does not require neither training nor tuning data.

The Test set contains 45,164 tokens and 1,022 total entities; it was created by adding tweets collected in December 2015 around the US primary elections and the Star Wars The Force Awakens Premiere.

The challenge evaluations are based on four metrics, but only three of them have to be considered for our case. Indeed, a metric was introduced for discriminating in case of tie in evaluation score, that is the *latency* but we have not such kind of problem in our experiments. The three main metrics are:

1. *strong_typed_mention_match (stmm)* that considers the micro average F_1 score on the annotations related to the mention extent and the type identification;

2. *strong_link_match (slm)* that considers the micro average F_1 score on the annotations related to the link for each mention;

3. *mention_ceaf (mc)* that considers the F_1 score for the NIL and not-NIL in the annotations.

The final score is computed considering such metrics, according to the formula: $score = 0.4 * mc + 0.3 * stmm + 0.3 * slm$. The scorer proposed for the TAC KBP 2014 task[2] was used to perform the evaluation.

5.3.2 Results and discussion

In Table 5.5 are reported the results of our system (highlighted in bold) compared with the best performances obtained by the participants to the #Micropost2016 workshop NEEL Challenge.

The state-of-the-art approach *ADEL* developed by (Plu et al., 2016) was used as the baseline. The last four columns report the micro average F_1 score on each of the three metrics taken into consideration, along with the the global *score* value. Particularly, the second column refers to the use of a supervised/unsupervised approach in each system. As it is shown, our system ranked

[2]available at `https://github.com/wikilinks/neleval/wiki/Evaluation`

Table 5.5: Comparation of QuASIt with the participants to #Micropost2016 NEEL Challenge

Rank	Approach	Team Name	F_1^{mc}	F_1^{stmm}	F_1^{slm}	*score*
1	sup	kea	0.641	0.473	0.501	0.5486
2	**unsup**	**QuASIt4NEEL**	**0.616**	**0.515**	**0.406**	**0.5227**
3	unsup	insight-centre @ nuig	0.621	0.246	0.202	0.3828
4	sup	mit lincoln laboratory	0.366	0.319	0.396	0.3609
5	sup	ju team	0.467	0.312	0.248	0.3548
6	sup	unimib	0.203	0.267	0.162	0.3353
*	*sup*	*adel*	*0.69*	*0.61*	*0.536*	*0.6198*

second, compared to the challenge participants, and this is a very remarkable result if we consider that our system has a performance very close to *kea* but we make use of an unsupervised strategy, while both *kea* and *ADEL* use a supervised one. Our systems performs similarly to *kea* in the *mc* measure, and outperforms it in the *stmm*, while has a decay in the *slm* that can be observed also in *ADEL* and *kea*.

Chapter 6

Conclusions

In this dissertation a novel cognitive model for Human-Machine Natural Language Interfaces is proposed along with its implementation. The main aim of this research was the realization of an architecture dedicated specifically to processing natural language using symbolic techniques of semantic inference, that could be able to both understand and produce natural language utterances. The term *cognitive* derives from the procedural semantics theory, which states that the cognitive processes related to natural language are executed on two kinds of knowledge, that is the perceptually grounded knowledge of the world, and the linguistic one. The proposed approach attempts to reproduce such processes in an artificial agent to make it able to both understand the query and produce the answer.

The main motivations of this work fall in the will to investigate the issues related to the inferential communication system like natural language as opposed to pattern based communication systems. Natural language communication implies a sort of "smartness" aimed at understanding, processing and inferring the real information conveyed by an utterance. On the other hand, in the semantic web scenario a software agent is able to access a huge amount of information in a structured manner. The information stored using ontological representation and its formal language enables the retrieval by means of automated systems. Ontology have been designed to capture the semantic knowledge of a domain in a machine understandable form. Current standards for ontologies managing, like OWL, are lacking in linguistic grounding, and are not able to achieve a clear link with natural language.

Moving from these premises, this work presents the attempt to model an artificial agent that is able to run proper cognitive processes to achieve correct understanding and production of natural language sentences. The agent runs these processes on its inner domain representation using both the linguistic knowledge and the domain knowledge represented in form of ontology. In this sense the QuASIt cognitive architecture, which is described in this dissertation, is both a rule-based and ontology-based natural language interface, able to implement a dialogic agent that answers to open-domain questions.

The model attempts to solve some limitations of both traditional ontology-based and statistical approaches, such as the small scale of the underlying natural language models. Ontology-based QA systems have more advantages than statistical ones: they offer additional information about the answer, provide reliability measures for their performance, and can motivate how the answer was produced. However, the linguistic models on which these methods rely on are poorly sensible to the evolution of language that is its *fluidity*: this is a typical aspect of human interactions. Moreover, such models fail when lexical resources are no available exhaustively for the language. Indeed, here was considered the case of language corpora that are not so widespread in the NLP literature, such as the Italian.

In the model, rules are aimed at understanding the query in terms of the linguistic typology of the question, and enabling its semantic processing as regards the search for the answer in the structured knowledge. Also the free explicatory text in support of the query is analyzed if available. The cognitive architecture reported in this dissertation keeps the ontology-based QA advantages, attempts to be open-domain like statistical approaches, and it is sensible to the fluidity of the languages.

The knowledge about the world is modeled by the domain ontology. Considering that the ontology can be replaced (the agent makes inferences based on the mere ontological structures) QuASIt can be considered an open-domain system.

The linguistic knowledge is the grammar of the language, and it is modeled by the usage patterns in that language, such as a word, a combination of words, an idiom, and so on. The generalization of the linguistic model, that is separated from the domain knowledge, is one of the main focuses of this work, and it is obtained by the *construction grammar* (CxG). Particularly, the Fluid Construction Grammar (FCG) has been used in this work because it is the unique computational framework reported in the literature.

The language model abstraction is the result of both the *continuum* and the *abstract categorizations* of constructions. The continuum is between quite abstract grammatical constructions and the *item-based* constructions. The abstract categorizations relate semantics to syntax and allow to conceptualize the meaning and the function of a language.

Modeling the linguistic knowledge by construction grammar allows to define general usage patterns of queries that represent the question's linguistic typology. The cognitive processes of query understanding are related to the operations performed by the agent for handling the pattern in the comprehension of questions. The production of the correct answer is obtained by other cognitive processes that, considering the content fitted to the query pattern, extract a subgraph from an ontology and make reasoning on the nodes of this subgraph for retrieving the correct information; such processes are based on a set of correspondences we defined purposely between the query's typology and some specific ontology properties.

The various applications of QuASIt described here reveal how the model was able to accomplish several tasks. The proposed approach makes QuASIt able to switch from a language domain to another without any training phase. Moreover, the rules for answer production consider two cases: essay questions, and multiple choice ones. In the first case the paragraphs extracted from the text contained in the nodes of the ontology subgraph that had been devised in the understanding phase, are analyzed for producing the correct answer by matching them against the retrieved properties in the ontology. In case of multiple choice questions, the candidate answers are used both to condition the pruning of the ontology subgraph and to guide the search inside the text contained in its nodes.

The use of similarity measures in the implementation of the underlying cognitive processes demonstrates their effectiveness also in detection and disambiguation of entities in informal language text messages.

The experiments carried out report satisfactory results. The system reaches performances above other state of the art tools using unsupervised approaches and dealing with more than one language, demonstrating the effectiveness of our approach in open domain, multi-language contexts. This results bring to consider QuASIt a state of the art system with respect to the performance values reported by the current literature.

The innovative research contribution can be resumed in the following points:

- Flexible coupling of the ontological and language systems
- Subcategorization and role identification in sentence structure
- Ambiguities resolution

The future works will be aimed at deepening the linguistic analysis of both the question and the answer either to refine the similarity measure, and to produce better answers through an improved definition of the FCG constructions. On this front, more complex phrasal structures will be analyzed. On the informal language side, future works will be devoted to devise more semantic information about the affective aspect of a message like the mood emerging in a social media discussion thread.

Bibliography

Ricardo Baeza-Yates, Berthier Ribeiro-Neto, et al. *Modern information retrieval*, volume 463. ACM press New York, 1999.

Collin F Baker, Charles J Fillmore, and John B Lowe. The berkeley framenet project. In *Proceedings of the 36th Annual Meeting of the Association for Computational Linguistics and 17th International Conference on Computational Linguistics-Volume 1*, pages 86–90. Association for Computational Linguistics, 1998.

Roberto Basili, Maria Teresa Pazienza, and Fabio Massimo Zanzotto. Exploiting the feature vector model for learning linguistic representations of relational concepts. In *Workshop on Adaptive Text Extraction and Mining (ATEM 2003) held in conjuction with European Conference on Machine Learning (ECML 2003)*, 2003.

Roberto Basili, Dorte H Hansen, Patrizia Paggio, Maria Teresa Pazienza, and Fabio Massimo Zanzotto. Ontological resources and question answering. In *Proceedings of the Workshop on Pragmatics of Question Answering at HLT-NAACL 2004*, 2004.

Richard Beaufort, Sophie Roekhaut, Louise-AmÃ©lie Cougnon, and CÃ©drick Fairon. A hybrid rule/model-based finite-state framework for normalizing sms messages. In Jan Hajic, Sandra Carberry, and Stephen Clark, editors, *ACL*, pages 770–779. The Association for Computer Linguistics, 2010. ISBN 978-1-932432-67-1. URL `http://dblp.uni-trier.de/db/conf/acl/acl2010.html#BeaufortRCF10`.

Benjamin K. Bergen and Nancy Chang. Embodied construction grammar in simulation-based language understanding. Technical report, EDS): CONSTRUCTION GRAMMAR(S): COGNITIVE AND CROSS-LANGUAGE DIMENSIONS. JOHN BENJAMIN PUBL CY, 2003.

Johan Bos. A survey of computational semantics: Representation, inference and knowledge in wide-coverage text understanding. *Language and Linguistics Compass*, 5(6):336–366, 2011.

Djallel Eddine Boubiche, Faouzi Hidoussi, Homero Toral Cruz, Abdelghani Bouziane, Djelloul Bouchiha, Noureddine Doumi, and Mimoun Malki. International conference on advanced wireless information and communication technologies (awict 2015) question answering systems: Survey and trends. *Procedia Computer Science*, 73:366 – 375, 2015. ISSN 1877-0509. doi: http://dx.doi.org/10.1016/j.procs.2015.12.005. URL `http://www.sciencedirect.com/science/article/pii/S1877050915034663`.

Paul Buitelaar, Philipp Cimiano, Peter Haase, and Michael Sintek. Towards linguistically grounded ontologies. *The semantic web: research and applications*, pages 111–125, 2009.

Annalina Caputo, Marco de Gemmis, Pasquale Lops, Francesco Lovecchio, Vito Manzari, and Acquedotto Pugliese AQP Spa. Overview of the evalita 2016 question answering for frequently asked questions (qa4faq) task. In *CLiC-it/EVALITA*, 2016.

Noam Chomsky. Aspects of the theory of syntax. 1965.

W. Croft. *Syntactic categories and grammatical relations: the cognitive organization of information.* University of Chicago Press, 1991. ISBN 9780226120904. URL `http://books.google.it/books?id=h1gLdOkH1GgC`.

William Croft and Alan D. Cruse. Polysemy: the construal of sense boundaries. In *Cognitive Linguistics*, chapter 5, pages 109–140. Cambridge University Press, 2004.

Aba-Sah Dadzie, Daniel Preotiuc-Pietro, Danica Radovanovic, Amparo Elizabeth Cano Basave, and Katrin Weller, editors. *Proceedings of the 6th Workshop on 'Making Sense of Microposts' co-located with the 25th International World Wide Web Conference (WWW 2016), Montréal, Canada, April 11, 2016*, volume 1691 of *CEUR Workshop Proceedings*, 2016. CEUR-WS.org. URL `http://ceur-ws.org/Vol-1691`.

Danica Damljanovic, Valentin Tablan, and Kalina Bontcheva. A text-based query interface to owl ontologies. In *LREC*. Citeseer, 2008.

Danica Damljanovic, Milan Agatonovic, and Hamish Cunningham. Freya: An interactive way of querying linked data using natural language. In *ESWC Workshops*, volume 7117, pages 125–138. Springer, 2011.

Adriel Dean-Hall, Charles L. A. Clarke, Jaap Kamps, Julia Kiseleva, and Ellen M. Voorhees. Overview of the trec 2015 contextual suggestion track. In Ellen M. Voorhees and Angela Ellis, editors, *TREC*, volume Special Publication 500-319. National Institute of Standards and Technology (NIST), 2015.

Leon Derczynski, Diana Maynard, Giuseppe Rizzo, Marieke van Erp, Genevieve Gorrell, Raphaël Troncy, Johann Petrak, and Kalina Bontcheva. Analysis of named entity recognition and linking for tweets. *Information Processing & Management*, 51(2):32–49, 2015.

Dominique Estival, Chris Nowak, and Andrew Zschorn. Towards ontology-based natural language processing. In *Proceeedings of the Workshop on NLP and XML (NLPXML-2004): RDF/RDFS and OWL in Language Technology*, pages 59–66. Association for Computational Linguistics, 2004.

Christiane Fellbaum. *WordNet*. Wiley Online Library, 1998a.

Christiane Fellbaum. *WordNet: An Electronic Lexical Database*. Bradford Books, 1998b.

Christiane Fellbaum, editor. *WordNet An Electronic Lexical Database*. The MIT Press, Cambridge, MA ; London, May 1998c. ISBN 978-0-262-06197-1.

Olivier Ferret, Brigitte Grau, Martine Hurault-Plantet, Gabriel Illouz, Laura Monceaux, Isabelle Robba, and Anne Vilnat. Finding an answer based on the recognition of the question focus. In *TREC*, 2001.

Charles Fillmore, Paul Kay, and Catherine O'Connor. Regularity and idiomaticity in grammatical constructions: The case of let alone. *Language*, 64:501–538, 1988.

Charles J Fillmore. Syntactic intrusions and the notion of grammatical construction. In *Annual Meeting of the Berkeley Linguistics Society*, volume 11, pages 73–86, 1985.

Pamela Forner, Jussi Karlgren, and Christa Womser-Hacker, editors. *CLEF 2012 Evaluation Labs and Workshop, Online Working Notes, Rome, Italy, September 17-20, 2012*, volume 1178 of *CEUR Workshop Proceedings*, 2014.

CEUR-WS.org. ISBN 978-88-904810-3-1. URL `http://ceur-ws.org/Vol-1178`.

Dirk Geeraerts and Hubert Cuyckens. Introducing cognitive linguistics. 2007.

Ben Goertzel, Ruiting Lian, Itamar Arel, Hugo De Garis, and Shuo Chen. A world survey of artificial brain projects, part ii: Biologically inspired cognitive architectures. *Neurocomputing*, 74(1):30–49, 2010.

Adele Goldberg and Laura Suttle. Construction grammar. *Wiley Interdisciplinary Reviews: Cognitive Science*, 1(4):468–477, 2010. ISSN 1939-5086. doi: 10.1002/wcs.22. URL `http://dx.doi.org/10.1002/wcs.22`.

Adele E Goldberg. *Constructions: A construction grammar approach to argument structure*. University of Chicago Press, 1995.

Adele E Goldberg. Constructions: a new theoretical approach to language. *Trends in cognitive sciences*, 7(5):219–224, 2003.

M. B. Habib and M. van Keulen. In *Proceedings of the System Demonstrations of The 53rd Annual Meeting of the Association for Computational Linguistics (ACL 2015), Beijing, China*, Beijing, China, July 2015. The Association for Computer Linguistics.

M.B. Habib and M. van Keulen. A generic open world named entity disambiguation approach for tweets. In *5th International Conference on Knowledge Discovery and Information Retrieval, KDIR 2013*. SciTePress, September 2013. URL `http://doc.utwente.nl/86471/`.

Bo Han and Timothy Baldwin. Lexical normalisation of short text messages: Makn sens a #twitter. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies -*

Volume 1, HLT '11, pages 368–378, Stroudsburg, PA, USA, 2011. Association for Computational Linguistics. ISBN 978-1-932432-87-9. URL `http://dl.acm.org/citation.cfm?id=2002472.2002520`.

Thomas Hoffmann and Graeme Trousdale. *The Oxford handbook of construction grammar*. Oxford University Press, 2013.

Thomas Hoffmann, Graeme Trousdale, and Hans C. Boas. Cognitive construction grammar, 2013.

Wesley A. Hoover and Philip B. Gough. The simple view of reading. *Reading and Writing*, 2(2):127–160, 1990. ISSN 1573-0905. doi: 10.1007/BF00401799. URL `http://dx.doi.org/10.1007/BF00401799`.

Carlos A Hurtado, Alexandra Poulovassilis, and Peter T Wood. Ranking approximate answers to semantic web queries. In *European Semantic Web Conference*, pages 263–277. Springer, 2009.

Ray S. Jackendoff. *Semantic Structures*. Cambridge: MIT Press, 1990.

Daniel Jurafsky and James H Martin. Speech and language processing: an introduction to natural processing computational linguistics, and speech recognition, 2000.

Esther Kaufmann and Abraham Bernstein. How useful are natural language interfaces to the semantic web for casual end-users? In *The Semantic Web*, pages 281–294. Springer, 2007.

Esther Kaufmann, Abraham Bernstein, and Lorenz Fischer. Nlp-reduce: A naive but domainindependent natural language interface for querying ontologies. In *4th European Semantic Web Conference ESWC*, pages 1–2, 2007.

Max Kaufmann and Jugal Kalita. Syntactic normalization of Twitter messages. In *International conference on natural language processing, Kharagpur, India*, 2010.

Paul Kay and Charles J. Fillmore. Grammatical constructions and linguistic generalizations: The What's X Doing Y? construction. *Language*, 75(1):1–33, 1999.

Catherine Kobus, François Yvon, and Géraldine Damnati. Normalizing sms: Are two metaphors better than one? In *Proceedings of the 22Nd International Conference on Computational Linguistics - Volume 1*, COLING '08, pages 441–448, Stroudsburg, PA, USA, 2008. Association for Computational Linguistics. ISBN 978-1-905593-44-6. URL `http://dl.acm.org/citation.cfm?id=1599081.1599137`.

G Lakoff. The invariance hypothesis: is abstract reason based on image-schemas? *Cognitive Linguistics*, 1(1):39–74, 1990.

George Lakoff. Women, fire, and dangerous things, 1987.

Ronald W Langacker. *Foundations of cognitive grammar: Theoretical prerequisites*, volume 1. Stanford university press, 1987.

Ronald W. Langacker. *Grammar and Conceptualization.* Walter De Gruyter, 1999. URL `http://www.isrl.uiuc.edu/~amag/langev/paper/langacker99grammarAnd.html`. Cognitive Linguistics Research, 14.

Jens Lehmann and Lorenz Bühmann. Autosparql: Let users query your knowledge base. *The Semantic Web: Research and Applications*, pages 63–79, 2011.

Chenliang Li, Jianshu Weng, Qi He, Yuxia Yao, Anwitaman Datta, Aixin Sun, and Bu-Sung Lee. Twiner: Named entity recognition in targeted twitter stream. In *Proceedings of the 35th International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '12, pages 721–730, New York, NY, USA, 2012. ACM. ISBN 978-1-4503-1472-5. doi: 10.1145/2348283.2348380. URL `http://doi.acm.org/10.1145/2348283.2348380`.

Fei Liu, Fuliang Weng, Bingqing Wang, and Yang Liu. Insertion, deletion, or substitution? normalizing text messages without pre-categorization nor supervision.

Vanessa Lopez, Michele Pasin, and Enrico Motta. Aqualog: An ontology-portable question answering system for the semantic web. *The Semantic Web: Research and Applications*, pages 135–166, 2005.

D. L. McGuinness. Question answering on the semantic web. *IEEE Intelligent Systems*, (1):82–85, Jan 2004. ISSN 1541-1672. doi: 10.1109/MIS.2004.1265890.

Deborah L McGuinness, Frank Van Harmelen, et al. Owl web ontology language overview. *W3C recommendation*, 10(10):2004, 2004.

Joel Nothman, Nicky Ringland, Will Radford, Tara Murphy, and James R. Curran. Learning multilingual named entity recognition from wikipedia. *Artif. Intell.*, 194:151–175, January 2013. ISSN 0004-3702. doi: 10.1016/j.artint.2012.03.006. URL `http://dx.doi.org/10.1016/j.artint.2012.03.006`.

Anselmo Peñas, Eduard H. Hovy, Pamela Forner, Álvaro Rodrigo, Richard F. E. Sutcliffe, Caroline Sporleder, Corina Forascu, Yassine Benajiba, and Petya Osenova. Overview of QA4MRE at CLEF 2012: Question answering for machine reading evaluation. In Forner et al. (2014). ISBN 978-88-904810-3-1.

URL http://ceur-ws.org/Vol-1178/CLEF2012wn-QA4MRE-PenasEt2012.pdf.

Emanuele Pianta, Luisa Bentivogli, and Christian Girardi. Multiwordnet: developing an aligned multilingual database. In *Proceedings of the First International Conference on Global WordNet*, January 2002. URL http://multiwordnet.fbk.eu/paper/MWN-India-published.pdf.

Arianna Pipitone and Roberto Pirrone. Cognitive linguistics as the underlying framework for semantic annotation. In *Semantic Computing (ICSC), 2012 IEEE Sixth International Conference on*, pages 52–59. IEEE, 2012.

Arianna Pipitone and Roberto Pirrone. A hidden markov model for automatic generation of er diagrams from owl ontology. In *ICSC 2014*, pages 135–142. IEEE Computer Society, 2014.

Arianna Pipitone, Maria Carmela Campisi, and Roberto Pirrone. An A* based semantic tokenizer for increasing the performance of semantic applications. In *2013 IEEE Seventh International Conference on Semantic Computing, Irvine, CA, USA, September 16-18, 2013*, pages 393–394, 2013. doi: 10.1109/ICSC.2013.75. URL https://doi.org/10.1109/ICSC.2013.75.

Arianna Pipitone, Vincenzo Cannella, and Roberto Pirrone. I-ChatbIT: an Intelligent Chatbot for the Italian Language. In Roberto Basile, Alessandro Lenci, and Bernardo Magnini, editors, *Proceedings of the First Italian Conference on Computational Linguistics CLiC-it 2014*. Pisa University Press, 2014.

Arianna Pipitone, Francesca Anastasio, and Roberto Pirrone. Howerd: A hidden markov model for automatic owl-erd alignment. In *ICSC*, pages 477–

482. IEEE Computer Society, 2016a. ISBN 978-1-5090-0662-5. URL `http://dblp.uni-trier.de/db/conf/semco/icsc2016.html#PipitoneAP16a`.

Arianna Pipitone, Giuseppe Tirone, and Roberto Pirrone. Chilab4it system in the qa4faq competition. In *CLiC-it/EVALITA*, 2016b.

Arianna Pipitone, Giuseppe Tirone, and Roberto Pirrone. Quasit: A cognitive inspired approach to question answering for the italian language. In *AI*IA 2016 Advances in Artificial Intelligence*, pages 464–476. Springer International Publishing, 2016c.

Arianna Pipitone, Giuseppe Tirone, and Roberto Pirrone. Named entity recognition and linking in tweets based on linguistic similarity. In *Conference of the Italian Association for Artificial Intelligence*, pages 101–113. Springer International Publishing, 2017.

Julien Plu, Giuseppe Rizzo, and Raphaël Troncy. Enhancing entity linking by combining NER models. In Sack et al. (2016), pages 17–32. ISBN 978-3-319-46564-7. doi: 10.1007/978-3-319-46565-4_2. URL `https://doi.org/10.1007/978-3-319-46565-4_2`.

Carl Pollard. *Lectures on the foundations of Hpsg*. Ohio State University, 1997.

Carl Pollard and Ivan Sag. *Head-Driven Phrase Structure Grammar*. Chicago University Press, Chicago, Illinois, 1994.

Borislav Popov, Atanas Kiryakov, Angel Kirilov, Dimitar Manov, Damyan Ognyanoff, and Miroslav Goranov. Kim–semantic annotation platform. In *International Semantic Web Conference*, pages 834–849. Springer, 2003.

Ellen Riloff. Information extraction as a stepping stone toward story understanding. *Understanding language understanding: Computational models of reading*, pages 435–460, 1999.

Alan Ritter, Sam Clark, Mausam, and Oren Etzioni. Named entity recognition in tweets: An experimental study. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, EMNLP '11, pages 1524–1534, Stroudsburg, PA, USA, 2011. Association for Computational Linguistics. ISBN 978-1-937284-11-4. URL `http://dl.acm.org/citation.cfm?id=2145432.2145595`.

Giuseppe Rizzo, Marieke van Erp, Julien Plu, and Raphaël Troncy. Making sense of microposts (#microposts2016) named entity recognition and linking (NEEL) challenge. In Dadzie et al. (2016), pages 50–59. URL `http://ceur-ws.org/Vol-1691/microposts2016_neel-challenge-report/`.

William H. Rupley, Timothy R. Blair, and William D. Nichols. Effective reading instruction for struggling readers: The role of direct/explicit teaching. *Reading & Writing Quarterly*, 25(2-3):125–138, 2009. doi: 10.1080/10573560802683523. URL `http://dx.doi.org/10.1080/10573560802683523`.

Stuart J Russell and Peter Norvig. Artificial intelligence: a modern approach (international edition). 2002.

Harald Sack, Stefan Dietze, Anna Tordai, and Christoph Lange, editors. *Semantic Web Challenges - Third SemWebEval Challenge at ESWC 2016, Heraklion, Crete, Greece, May 29 - June 2, 2016, Revised Selected Papers*, volume 641 of *Communications in Computer and Information Science*, 2016. Springer. ISBN 978-3-319-46564-7. doi: 10.1007/978-3-319-46565-4. URL `https://doi.org/10.1007/978-3-319-46565-4`.

Ivan A Sag, Thomas Wasow, Emily M Bender, and Ivan A Sag. *Syntactic theory: A formal introduction*, volume 92. Center for the Study of Language and Information Stanford, CA, 1999.

Ivan A. Sag, Thomas Wasow, and Emily Bender. *Syntactic Theory: A Formal Introduction*. Center for the Study of Language and Information, Stanford, 2nd edition, 2003. URL `http://csli-publications.stanford.edu/site/1575864002.html`.

Philippe Smets. Belief functions: the disjunctive rule of combination and the generalized bayesian theorem. In *Classic Works of the Dempster-Shafer Theory of Belief Functions*, pages 633–664. Springer, 2008.

Dan Sperber and Deirdre Wilson. Precis of relevance: Communication and cognition. *Behavioral and brain sciences*, 10(4):697–710, 1987.

Michael Spranger, Simon Pauw, Martin Loetzsch, and Luc Steels. *Open-ended Procedural Semantics*, pages 153–172. Springer US, Boston, MA, 2012. ISBN 978-1-4614-3064-3. doi: 10.1007/978-1-4614-3064-3_8. URL `http://dx.doi.org/10.1007/978-1-4614-3064-3_8`.

Guy L. Steele, Jr. *Common LISP: the language (2nd ed.)*. Digital Press, Newton, MA, USA, 1990. ISBN 1-55558-041-6.

Luc Steels. Introducing fluid construction grammar, 2011a.

Luc Steels. *Design patterns in fluid construction grammar*, volume 11. John Benjamins Publishing, 2011b.

Luc Steels. *The talking heads experiment: origins of words and meanings*, volume 1. Language Science Press, 2015.

Luc Steels and Joachim De Beule. Unify and merge in fluid construction grammar. In P. Vogt and et al., editors, *Symbol Grounding and Beyond: Proceedings of the Third International Workshop on the Emergence and Evolution of Linguistic Communication*, Lecture Notes in Computer Science, pages 197–223. Springer, 2006. doi: 10.1007/11880172_16. URL `http://www.isrl.uiuc.edu/~amag/langev/paper/Steels06Unify_merge_FCG.html`.

Luc Steels and Joachim De Beule. A (very) brief introduction to fluid construction grammar. In *Proceedings of the Third Workshop on Scalable Natural Language Understanding*, pages 73–80. Association for Computational Linguistics, 2006.

Lappoon R Tang and Raymond J Mooney. Using multiple clause constructors in inductive logic programming for semantic parsing. In *ECML*, volume 1, pages 466–477. Springer, 2001.

Thanh Tran, Philipp Cimiano, Sebastian Rudolph, and Rudi Studer. Ontology-based interpretation of keywords for semantic search. *The Semantic Web*, pages 523–536, 2007.

Christina Unger, Lorenz Bühmann, Jens Lehmann, Axel-Cyrille Ngonga Ngomo, Daniel Gerber, and Philipp Cimiano. Template-based question answering over rdf data. In *Proceedings of the 21st international conference on World Wide Web*, pages 639–648. ACM, 2012.

Aobo Wang, Tao Chen, and Min-Yen Kan. Re-tweeting from a linguistic perspective. In *Proceedings of the Second Workshop on Language in Social Media*, LSM '12, pages 46–55, Stroudsburg, PA, USA, 2012. Association for Computational Linguistics. URL `http://dl.acm.org/citation.cfm?id=2390374.2390380`.

William E. Winkler. String comparator metrics and enhanced decisionn rules in the fellegi-sunter model of record linkage. In *Proceedings of the Section on Survey Research*, pages 354–359, 1990.

Valentin Zhikov, Laura Tolosi, Petya Osenova, Kiril Simov, and Georgi Georgiev. Cross-lingual answer validation.

Qi Zhou, Chong Wang, Miao Xiong, Haofen Wang, and Yong Yu. Spark: adapting keyword query to semantic search. In *The Semantic Web*, pages 694–707. Springer, 2007.